

3-2019

ABILITY SPORTS AND EDUCATION FESTIVAL MOBILE APPLICATION

Thomas Saldana

Follow this and additional works at: <https://scholarworks.lib.csusb.edu/etd>



Part of the [Computer Engineering Commons](#)

Recommended Citation

Saldana, Thomas, "ABILITY SPORTS AND EDUCATION FESTIVAL MOBILE APPLICATION" (2019).
Electronic Theses, Projects, and Dissertations. 942.
<https://scholarworks.lib.csusb.edu/etd/942>

This Project is brought to you for free and open access by the Office of Graduate Studies at CSUSB ScholarWorks. It has been accepted for inclusion in Electronic Theses, Projects, and Dissertations by an authorized administrator of CSUSB ScholarWorks. For more information, please contact scholarworks@csusb.edu.

ABILITY SPORTS & EDUCATION FESTIVAL MOBILE APPLICATION

A Project
Presented to the
Faculty of
JHB College of Business and Public Administration,
California State University
San Bernardino

In Partial Fulfilment
of the Requirements for the Degree
Master of Information Systems and Technology

by
Thomas Saldana

March 2019

ABILITY SPORTS & EDUCATION FESTIVAL MOBILE APPLICATION

A Project
Presented to the
Faculty of
JHB College of Business and Public Administration,
California State University
San Bernardino

by
Thomas Saldana
March 2019

Approved by:

Conrad Shayo, PhD., Chair

Date

Frank Lin, PhD, Reader

Jay Varzandeh, PhD., Chair, Information & Decision Sciences Department

Abstract

The purpose of this project is to create an iOS native mobile application for the Ability Sports & Education Festival at the California State University of San Bernardino. The iOS application will solve three business needs. First, it will allow participants to view the events, hours, schedules, coaches, and resources of the Ability Sports and Education Festival. Second, it will allow participants to route, search, and find event locations using ArcGIS. Lastly, it will allow participants to register for the event right from their phones.

Keywords: iOS, mobile, application, ArcGIS, Ability Sports & Education Festival

Dedication

Dedicated to my family & to all the people that have helped me to achieve my accomplishments

Contents

1. Introduction
2. Learning about the problem
3. The Proposed Solution
4. Application of UML/UP Process
5. Requirements & Analysis & Effort Estimation
6. Design and Implementation
7. Testing and Deployment
8. Summary
9. References
10. Appendices.

Ability Sports & Education Festival Mobile Application

Every year the California State University of San Bernardino hosts a disability sports festival in October for people with disabilities, injuries, or illnesses. The event offers people with disabilities, community members, and volunteers a chance to compete in sports designed for people with disabilities. The event has been hosted at California State University, San Bernardino for the last 12 years, and each year hundreds of participants attend the event. Every year the event grows, and this year, more than 1000 people are expected to attend the event.

The event is similar to the Paralympics but geared more toward community members. During the event, participants are encouraged to participate in all types of sports such as adaptive swimming, archery, wheelchair basketball, track and field, and rock climbing. It is a great event that encourages people with disabilities to go outside their comfort zones and try new sports and things they never could have imagined doing. Not only that, but it also helps people with disabilities discover new talents and skills they never knew they had. The event changes the lives of the participants, and it is all about encouraging people with disabilities to try new things and showing them that they can do anything any able body can.

Overview. The purpose of this project is to design and develop an iOS Mobile application for the Ability Sports & Education Festival that is held every October at the California State University, San Bernardino. Although my primary goal for this project will be to design, test, and develop the iOS version of the Ability Sports & Education Festival application, I will also be helping in the design, testing and deployment of the Android version of the application. I will use the skills I learned in SCM 515 (Project Management), IST 645

(Information Systems Analysis and Design), IST 646 (Information Systems Planning, Strategy, and Policy), IST 647 (Information Based Management), and some of my other master courses to successfully complete this project. The iOS application will be developed using the Swift object-oriented programming (OOP) language and the XCode 10.1 integrated development environment (IDE). The Android application will be developed using the Java OOP language and the Android Studio IDE. Both applications will use the same back end infrastructure and both applications will integrate third party API's to send or retrieve data, display maps, and register users. Additionally, they will both have access to real time databases; they will both integrate ArcGIS, and they will both connect to Qualtrics—the backend registration system.

Goals. The primary goal of the University, the Service for Students with Disabilities, and this project is to create an iOS and Android application for the Ability Sports & Education Festival. Additionally, the goal is to also create an appealing application user interface that is easy to use and accessible for students with disabilities.

Objective. The objective of this project and the Ability Festival application is to use information and Mobile technologies to better assist the Service for Students with Disabilities program and the Ability Sports and Education Festival.

Learning about the problem

Personally, I did not know this event existed at the California State University of San Bernardino. I found out about this event when I was looking for a master's culminating project to work on. That is when I met Ginger Hartman, the director of the Ability Sports & Education

Festival. She mentioned to me that she would like to develop a Mobile application for the event. Since my back ground is in computer science and I have experience in Mobile application development, ArcGIS and systems administration, I thought it would be a great project to work on. I agreed to help her design, develop, plan, and manage the first stages of the iOS native Mobile application as my final culminating project. I chose this as my culminating project because I thought it was a unique project that had would help me finish my master's program and at the same time be beneficial to the event, participants and the university. In a way, this project is an opportunity for me to give back to my University.

The Business Need for An Application. The Ability Sports & Education Festival has been held at the California State University, San Bernardino for more than a decade. The event registers hundreds of users. It has many sports, and it offers a lot of services throughout the event. However, currently participants have to register on paper or on a website, which makes the registration process take hours. Additionally, participants who are not familiar with the California State University of State San Bernardino campus get lost and have a hard time finding sport clinics and other events. Lastly, participants miss out on sports clinics and other clinics because they don't have an easy way of seeing everything the event has to offer. A Mobile application would fix all that. Although every year the event is successful, the registration process and the overall experience of the event can be improved with the help of a Mobile application. With the help of a Mobile application, participants would be able to register right from their phones on the day of the event, or even the night before, without having to wait in long lines. Additionally, participants would be able to find clinics much faster, and participants would be able to use dynamic ArcGIS maps to find routes, sport clinics and other event

locations, all from their smart phones. They would not have to use a paper map to orient themselves, and all information about the event would be right on their phones.

Why hasn't this problem been addressed? iOS and Android applications can cost thousands or even hundreds of thousands of dollars to develop, depending on their complexity and who is developing them. Additionally, it is not easy to find people who can develop iOS or Android applications and are available at a cheap cost. For this reason, the Ability Sports & Education Festival iOS and Android applications have never been developed. The school and the department that oversees the festival just don't have the budget to create such complex applications for a non-profit program.

Additionally, currently, there is no application out there that can do what this application requires. This application is too unique, and it does not exist in the Apple App Store nor the Google Play Store.

Proposed Solution. The proposed solution is to create an application for the event that would allow participants to register, find locations, and over all find sports clinics and other services on the day of the event. Overall, like mentioned above, the application would help participants save time during registration, and it would help participants have a better experience during the event.

Procedure. The application will be developed for both the iOS and Android operating systems. The Android version of the application will be developed by Mohamed Rahman, who is a senior computer-science student at Cal State San Bernardino. He will develop the application

using Java and the Android Studio IDE, and I will develop the iOS version of the application using Swift and XCode 10.1. However, since the application is also going to integrate ArcGIS, another student, Eric Niles, who is majoring in graphic design, will develop the ArcGIS map and geodatabase using ArcMap and ArcGIS Pro. After Eric creates the ArcGIS map and geodatabase, Mohamed and I will publish the map to the school's ArcGIS Enterprise server that is located in Azure, and we will integrate it into both applications using the ArcGIS Runtime SDK for Android and the ArcGIS Runtime SDK for iOS. The application's user interface which includes the design, colors, images, backgrounds, and overall user experience, will be designed entirely by the client, Ginger Hartman, and her graphics designer, Viridiana Guzman, who is a senior at Cal State San Bernardino and majoring in graphics design. Overall, since I have experience developing Mobile applications, I will also help manage all the different technical parts of this project.

Scope. There is a lot of risk in this project, not because it is hard but because there is a lack of developers and human resources needed to complete it. In the Mobile development world, it takes a team of around 3 to 4 experienced developers, working full time and writing an average of 50 lines of code (LOC) a day, an average of 4 to 6 months to develop a Mobile application. The actual time it takes to develop a Mobile application depends on many different factors, but it ultimately depends on the complexity of the application and the number of available human resources available to develop it.

Before I estimated the number of human resources needed for this project, I gathered functional and non-functional requirements from the client (which are discussed in the Appendix D of this document), and I scoped out the application using the methods, tools, and concepts I learned in

IST 645 (Information Systems and Design), SCM 515 (Project Management), and previous experiences.

The following are some of the things that I considered and asked the client before scoping out the project's complexity:

- Is the application small, medium, or large?
- Is the user interface simple, basic, or polished?
- Is the application static or dynamic?
- Does the application have user accounts and user profiles?
- Does the application allow searching, uploading and the sharing of information?
- Does the application display maps or locations?
- Does it synchronize with a cloud or does it integrate third party API's?
- How many databases will be needed to support the application?
- Does the application communicate with servers?
- Does the application have security or encryption?
- Is the application multi-platform—iOS and Android?

Based the effort estimation and risk analysis concepts that I learned in IST 645, my experience, and the client requirements, the Ability Sports & Education Festival application is going to be a complex application, and it is going to have a sophisticated back end. Technically, in order to build both the iOS and Android applications in a period of 6 months, this project would need 4 iOS developers, 4 Android developers, 1 ArcGIS cartographer, 1 - 2 UI/UX designer, 1 systems administrator/engineer, 1 cloud support engineer, and realistically, one technical project manager. However, the project only has 2 developers (one Android and one iOS), 1 ArcGIS

cartographer, 1 cloud-support engineer and 2 UI/UX designers. Currently, I have the most experience in the field of systems administration and Mobile application development, so I will play multiple roles in this project. I will be the iOS developer, the systems engineer that puts all the pieces of the project together, and the person who oversees the whole project from start to finish.

Overall, the project is do-able with the low number of human resources that we have, but it is going to take at least 1 year to complete, and that most likely, it is only going to include the front end of the application and the integration of ArcGIS, not user accounts or dynamic content.

However, the client is aware of this and has no problems with it. She expects this project to take anywhere from 1 to 2 years to complete.

Moreover, I should mention that although we do not have the right number of human resources to work on this project, we have everything we need to complete this project successfully. We have unlimited licenses for ArcGIS products. We have access to Azure and are able to host servers on the cloud at no additional cost since we are part of the University. We have MacBook's, iMacs, phones and development accounts for testing and development, and we have time since the client has not placed a hard deadline knowing that there is a limited number of human resources (see Appendix A for complete list of human resources).

Third Party API's. Because of the lack of technical resources and time, the Ability Sports & Education Festival application's backend which includes databases, servers, registration systems, etc. will not be developed from scratch. It would be extremely hard for students to develop a robust back end infrastructure in such a short period of time and with

limited skills and experience. For this reason, the application will integrate third party API's and other services for the backend infrastructure.

The following is an overview of the third-party API's and services that will be used to support both the iOS and Android applications.

ArcGIS. The most complex part of this project will be the integration of ArcGIS. ArcGIS is a geographic information system that is used to create maps and analyze spatial data. It is a platform that is developed by the Environmental Systems Research Institute (ESRI) in Redlands, California, and it consists of different software products such as ArcGIS Online, ArcGIS Enterprise, ArcMap, ArcGIS Pro, and ArcGIS Runtime SDKs.

The application will integrate the ArcGIS platform in both the iOS and Android applications using the ArcGIS Runtime SDK for Android and the ArcGIS Runtime SDK for iOS. The maps of the event will be created using ArcMap and ArcGIS Pro, and they will be hosted on an ArcGIS Enterprise server that is located in Azure and that is scalable and highly available.

Google Firebase. Firebase is a comprehensive Mobile development platform that allows application developers to build back-end services quickly, efficiently, and at a low cost. Firebase offers a lot of different services such authentication, hosting, cloud storage, and functions. However, for this application, Firebase will be used as a real-time database to create dynamic views that will allow for dynamic content in both the iOS and Android applications, and it will be used for authentication and to store user accounts and profiles.

Microsoft Azure. Similar to Google's Firebase service, Microsoft's Azure cloud will be used as part of the back-end infrastructure. Azure will be used to host the ArcGIS Enterprise servers that will host and serve the ArcGIS maps to both the Android and iOS applications on the day of the event.

Qualtrics. Qualtrics is the last piece of the backend infrastructure. Qualtrics is an experience management platform. It can be used both as a database to hold data and as a form to gather data. Qualtrics will be used to register participants on the day of the event, and it will be used to store all sorts of registration information such as first name, last name, date of birth, gender, and address.

Back-End Infrastructure–Over View. To visualize and draw the backend infrastructure of the application, I used Gliffy, which is a web site that is used to create UML, UP, and network diagrams. It is similar to Visio, and it is great for quick network diagrams. (see Appendix B for a diagram of the backend infrastructure needed to support the Ability Sports & Education Festival application).

Time Management. Time management is a really important part of this project. This project has many tasks. It is complex, and it has many pieces. Additionally, certain tasks precede other tasks, so creating a schedule was a must. I never really paid attention to planning and time management, but after taking multiple courses on planning during my masters, like the SCM 515 project management class, I realized the importance of scheduling in a project. Scheduling helps projects succeed. When we started this project back in January 2019, we did not really know

where to start, and we did not know who was going to be responsible for the different tasks in the project. After learning about Microsoft Project in one of my classes, what I decided to do as acting project managers was create a Microsoft Project to keep track of all the tasks needed to successfully complete the different phases of this project. I am not a Microsoft Project expert, but the skills that I learned in my masters were enough to help me create a Microsoft Project and use it efficiently to keep track of all the work that needed to be done (see Appendix C for the schedule and Gantt chart of the first phase of this project). I created the schedule based on past experiences and previous Mobile application development projects.

Regarding time, the official start date of this project was January 15th, 2019. That is when I started gathering requirements from the client and scoping out the project. Before speaking with the client, I thought the app was going to be static and used only one day out of the year, so I thought this project was only going to take 3 months, at most, to complete. However, this application is not going to be static, and it is going to integrate third party API's. Because of this, this project is going to take approximately 8 – 12 months to complete.

I will work on this project during the entire quarter of Winter 2019. However, I do not think that I will finish the project by the end of the quarter. I am estimating that I will finish two-thirds of the front end of the iOS and Android applications by the end of the Winter 2019 quarter.

Additionally, I am estimating that I will be able to integrate some of the ArcGIS features into the app by the end of the quarter as well.

After the quarter ends and I graduate, I plan to stay, help out, and finish the frontend of the application and the integration of the ArcGIS. My goal is to finish a working version of the iOS application by September 1st. I would really like for this application to be used during this year's event. I feel that if I leave the project right after I graduate it would be too hard for another

student developer to pick up where I left off. Before I leave this project, I also plan to train another computer science student who is younger, maybe a junior or a senior, so that he or she can continue to support the Ability Sports & Education Festival application after I am gone.

Application of UML/UP Process

There are many different types of software development and design techniques that I could have used in this project. However, for this project, I choose to use the UML/UP process to design and develop this application. The main reason I choose the UML/UP process is because I am familiar with it. Throughout my undergrad and graduate studies I always used the UML/UP process to design and develop software applications.

Methodology. In the SCM 515 class, I learned about the Agile methodology, which is a specific type of Rapid Application Development. However, in the beginning of this project, I used the Waterfall methodology. Because of this, it was very hard for me to determine exactly what frameworks, classes, or design patterns I needed to complete this project successfully. Quickly, around the fourth week of this quarter, when I was learning about Agile in SCM 515, I realized that I could not plan the entire project from start to finish because it was too dynamic, so around that time I decided to try the Agile methodology to develop and design the application more efficiently.

Agile worked for me. I was able to conceive the application through the functional and non-functional requirements. I was able to analyze and design the first stages of the application, and I was able to begin the construction phase around 5th week of the quarter.

I still need testing and more analysis and design, but I am happy with the work I have done, and the Agile Methodology has helped me by allowing me to divide and conquer the different components of the application and allow me to conceive, analyse, design and test the different components of this application over and over again in an efficient manner.

In the future, what I can do to improve the quality and management of the application, is spend more time in the analysis and design phases of the project.

Analysis. To discover potential candidate classes, I read over all functional and non-functional requirements. I looked at use case diagrams and use case specifications. Additionally, I also use the following method that underlines nouns and noun phrases to discover potential classes in a system description.

The following illustrates the potential nouns and noun phrases in the system description:

The **Ability Sports** & Education Festival system will allow participants to view **events**, **coaches**, **resources**, **sports** from a **home screen**. Additionally, the Ability Sports & Education Festival system will allow participants to **register** right from their phones.

Participants may **search** for **event locations** using a dynamic ArcGIS **map**, that will allow them to view **sports clinics**, **event boundaries**, **first-aid tents**, **water stations**, and other event resources. Additionally, participants will be able to **route** to different **locations** during the event.

Participants will be able to navigate anywhere in the app using a **home button**, and **side menu**.

Noun/Verb

The following is a list of nouns and verbs from both the system description above and use case specifications in Appendix I.

- Ability Festival
- Accounts
- Events
- Coaches
- Resources
- Menu
- News
- About US
- Register
- Sports
- Home Screen
- Register
- Side Menu
- Menu Table
- Map
- Set Initial Map
- Load Map
- Display Map
- Route
- Location
- Unique Value Render
- Identify Icon
- Search

Candidate Class List

- Menu
- Ability Festival

- Menu Table
- Home
- Events
- Coaches
- Map
- News
- About US
- Sports
- Registration
- Reveal Side Menu

Candidate Attributes

- Menu Button
- Home Button
- Side Menu Button
- Events Button
- Coaches Button
- Map Button
- Home Button
- Register Button
- Keynote Button
- Schedule
- Map View

Candidate Operations

- View Did Load
- Display Sports

- Table View
- Set Menu Selection
- Reveal Side Menu
- Go Home
- View Sports
- View Coaches
- Register
- Display Map
- Display Clinics

To view the preliminary and extended class diagrams, please see Appendix J.

Requirements

The Ability Sports & Education Festival iOS application will be simple to use, appealing to users, and at the same time offer a large amount of information. Like mentioned above, the Ability Sports & Education Festival system will use dynamic ArcGIS maps, and it will have dynamic content. All application requirements were requested by the client, Ginger Hartman, who oversees the Ability Sports & Education Festival (please see Appendix D to view the functional and non-functional requirements).

Use Case Specifications. For this project, I used Use Case Specifications to represent the application as a software system. Appendix E, F, and G detail use case specifications, use case models, and actor semantics.

Analysis & Effort Estimation

Like any type of project, software engineering projects have to be planned in a systematic way in order to prevent project failure. Watzlawick (2014) states that the motivation for effort estimation in software projects come from the fact that historically most software development projects:

- Take more time to complete than expected.
- Cost much more than expected
- Generate a product that does not have the quality expected.
- Generate a product that does not have the scope expected.

Furthermore, Watzlawick (2014) that software development teams and clients must learn to expect the right time, cost quality, and scope regarding their project and their team using effort estimation techniques.

Effort Estimation. In IST 645, I learned a number of different parametric and non-parametric effort estimation techniques. For example, one non-parametric effort estimation technique I learned is the Smith technique, which is a technique that basically asks a Project Manager, “Smith,” or a development team for feedback on how much time he or the team thinks will be needed to develop a system. Of course, the smith technique only works if the project manager or team developing a system has experience developing that type of system because someone who has no experience developing a system could never estimate, well enough, how long it would take to develop a system based on intuition.

Another two no-parametric effort estimation techniques I learned were the *estimation by analogy* and *pricing to win*. The estimation by analogy technique, according to Watzlawick (2014) “is in

fact the pragmatic basis for the expert judgement technique. It is assumed that the effort for developing a new system will be similar to the effort for developing other similar systems,” so experts can estimate the effort to develop a system based on previous experiences. The pricing-to-win effort estimation technique is a technique where the cost of a project is equal to the price that is believed will win the contract Wazlawick (2014). However, both effort estimation techniques have flaws. The estimation by analogy is only as good as the experience an estimator has, and the pricing to win effort estimation technique does not produce quality applications, since costs are restricted.

Apart from the non-parametric techniques I learned, I also learned about two parametric techniques: COCOMO II (CONstructive COst Model II), and Function points analysis.

COCOMO or the Constructive cost model, like the previous effort estimation techniques, is a regression model that is used to estimate the costs and efforts associated with a software project. However, it does it through actual calculations and up to 16 effort multipliers.

After looking and analyzing all parametric and non-parametric techniques, I choose COCOMO to calculate the effort needed to complete this project. Although COCOMO has a flaw in that the KSLOC has to be known before it can be used, I still decided to use it because I have developed similar applications, and I know that this application is going to be around 10,000 lines of code. In a way, I am using a mix of no-parametric and parametric techniques to estimate the effort needed to complete this application. I am going to use my experience, or Smith method, to estimate that this application is going to be around 10,000 lines of code, and I am going to use that estimate along with COCOMO to calculate the effort needed to complete this application.

Overall, I also choose COCOMO because it has three model types: basic (which is for small projects); intermediate (which is for intermediate size projects); and detailed (which is for large-complex projects). Additionally, COCOMO has three project types: *organic mode*, *semi-detached*, and *embedded*. *Organic mode* is for relatively small projects that contain anywhere between 0 and 50 KLOC and have a small number of developers. *Semi-detached* mode is for intermediate size projects that are more complex and require more skills. Semi-detached projects usually have anywhere between 50 KLOC and 300 KLOC. Lastly, *embedded* projects are a lot more complex, have complex constraints, and required more skilled developers to complete. Embedded projects usually are the large applications that have more than 300 KLOC.

For this project, I used the basic model along with the organic mode, since this project is not large and it is going to integrate third-party API's and will only require around 10,000 LOC.

Based on my estimate, I calculated the effort and development time for each of the three modes using the following equations:

$$E = a_b(KLOC)^{bb}$$

$$D = c_b(E)^{db}$$

And this is what I came up with using 10,000 LOC:

1) Organic Mode

$$E = 2.4 (10)^{1.05} = 26.92 \text{ PM}$$

$$D = 2.5 (26.92)^{0.38} = 8.75 \text{ M}$$

2) Semi detached Mode

$$E = 3.0 (10)^{1.12} = 39.54 \text{ PM}$$

$$D = 2.5 (39.54)^{0.35} = 9 \text{ M}$$

3) Embedded Mode

$$E = 3.6 (10)^{1.20} = 57.05 \text{ PM}$$

$$D = 2.5 (57.05)^{0.32} = 9.11 \text{ M}$$

I believe the above results are somewhat accurate. It would take around 4 - 6 months to complete this project with the right number of developers.

Costs. The reason Mobile applications cost a lot of money is because 90% of the costs associated with the development of an application go to the salaries of the developers and other technical employees. Technically, it does not cost much to develop an iOS application. All a developer needs is a laptop with the MacOS operating system, an Apple developer account, and his or her mind—that is all. However, because the people who develop Mobile applications are considered technical and usually get paid large salaries, Mobile applications cost a lot of money to develop.

The following hypothetical table shows why a simple-to-medium size iOS application that would take 5 months to develop can easily cost over 6 figures:

Title	Cost (Per Year)	Cost (For 5 months)
-------	-----------------	---------------------

iOS Developer	\$60,000	\$25,000
iOS Developer	\$60,000	\$25,000
iOS Developer	\$60,000	\$25,000
iOS Developer	\$60,000	\$25,000
UI/UX Designer	\$40,000	\$10,000
Cloud Support Engineer	\$70000	
Project Manager	\$85000	
Total:	\$435,000	\$110,000

Please note that I did not include the cost of a cloud support engineer, manager or any other possible technical person in a development team. Additionally, I did not include the salaries of the Android developers. The salaries of the Android developers would easily double the costs above. Moreover, most development companies are in the Mobile app development business to make money, so they charge additionally fees on top of employee salaries to make a profit as well. And lastly, I assume that developers make \$60,000 a year. However, that number can be much higher, depending on where an application is developed.

Even though the costs in the table above are high, both the iOS and Android Ability Festival & Education Festival applications will not cost anywhere near that. The primary reason is because students are developing these applications, and students are cheap. All students developing these applications make \$13.00 an hour, and they work either 20 hours a week or 10 hours a week—only.

The following table shows the actual costs this project:

Name of Employee	Title	Hourly Wage	Cost (For 9 months)
Mohamed Rahman	Android Developer	\$13.00	\$15,600
Thomas Saldana	iOS Developer	\$0.00	\$0.00
Eric Niles	ArcGIS Cartographer	\$13.00	\$4,680
Viridiana Guzman	UI/UX Designer	\$13.00	\$15,600
Ching Yi Want	Cloud Administrator	\$0.00	\$0.00
Danny Vasquez	Qualtrics Administrator	\$0.00	\$0.00
Total:			\$35,880

I am a volunteer, so I will not get payed. Additionally, Ching Yi Wang and Danny Vasquez are faculty, so they already get payed through the University.

Design and Implementation

After gathering requirements, I sat down with the client and the designer of the application, and I helped them come up with designs based on other popular Mobile applications such as Facebook, Instagram, Adobe, and Kaiser Permanente. After a few weeks of design, the client and the designer came up with the design of the application (see Appendix L to view the detailed designs of the application).

Testing and Deployment

Currently, as of March 15th 2019 and at this point in the project, we have developed two thirds of the front end of the iOS and Android applications, and we have begun to integrate ArcGIS into both applications. However, we are estimating that we still have 7 more months before we are done with version v1.0 of the application. We are hoping to finish the front end of the iOS application (which will include ArcGIS functionality) by September 1st, 2019, and we are hoping to finish the Android version of the application by September 15th, 2019.

Alpha Testing. We plan to test the application sometime at the end of next quarter (Spring 2019). What we will do is talk to potential users, faculty, staff, and volunteers to test the apps functionality. We will observe user's interactions with the app, and we will come up with performance tests. The quality and performance tests will However, since the application is still currently in the early phases of development, we did not perform alpha testing this quarter.

On Going Support. After the first release, we hope new student developers can learn the application so that they can support and expand it for years to come. However, we know that this will be a challenge, since it takes time to learn the application development process and ArcGIS. For this reason, we are currently leaving a lot of documentation behind, and we are documenting even the simplest of procedures so that future student can pick up right where we left off.

Issues in the Chosen Project's Performances. Overall, we have not had any performance issues, or other critical issues. The project is going as planned, everyone is on schedule, and we currently don't have any bugs in our code that are causing performance issues. However, I do foresee an issue on the day of the event. If the ArcGIS server can't handle the requests of hundreds or even thousands of users, then it most likely it will crash, and participants

will not be able to route or find locations using the ArcGIS maps. To prevent this, we will perform stress tests a couple of weeks before the go-live of the application, and we will aim to make the ArcGIS server highly available or scalable on the day of the event. The goal is to have an ArcGIS server that runs well and is able to serve the Ability Sports & Education Festival application continuously.

Summary

Overall, I learned a lot about what it takes to design an application that has many pieces and resources. I learned how hard it is to gather requirements and plan an application from start to finish, before the actual construction of the application begins. I also learned how hard it is to manage human resources. It is not easy communicating project requirements and deliverables with project team members, or the client and it is not easy keeping track of all the moving pieces of a project. However, with the help of the tools, concepts, and methods that I learned during my masters, I would say that I was able to manage the project successfully. I can see this project being completed by the end of this year, and I can see it being a successful tool that is going to help the participants of the Ability Sports & Education Festival enjoy the event even more for years to come.

References

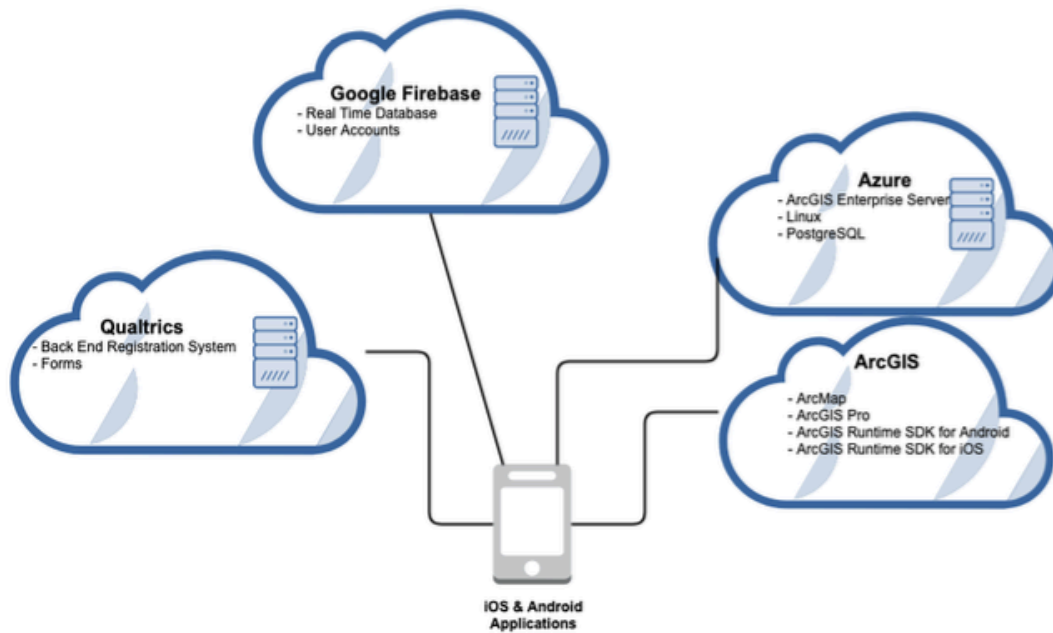
- “ArcMap.” *ArcMap* | *ArcGIS Desktop*, desktop.arcgis.com/en/arcmap/.
- “ArcGIS Enterprise.” *ArcGIS Enterprise* | *ArcGIS Enterprise*, enterprise.arcgis.com/en/.
- ArcGIS Runtime SDK for Android (latest). (2018). Retrieved from <https://developers.arcgis.com/android/latest/>
- ArcGIS Runtime SDK for Android (latest). (n.d.). Retrieved from <https://developers.arcgis.com/android/latest/>
- Madachy, R. (2018). Retrieved from <http://csse.usc.edu/tools/COCOMOII.php>
- “Microsoft Azure Cloud Computing Platform & Services.” *Microsoft Azure Cloud Computing Platform & Services*, azure.microsoft.com/en-us/.
- Google, Google, firebase.google.com/.
- Software Engineering | COCOMO Model. (2018, April 13). Retrieved from <https://www.geeksforgeeks.org/software-engineering-cocomo-model/>
- Wazlawick, R. S. (2014). *Object-Oriented Analysis and Design for Information Systems*. Waltham: Elsevier Science & Technology.

Appendix A

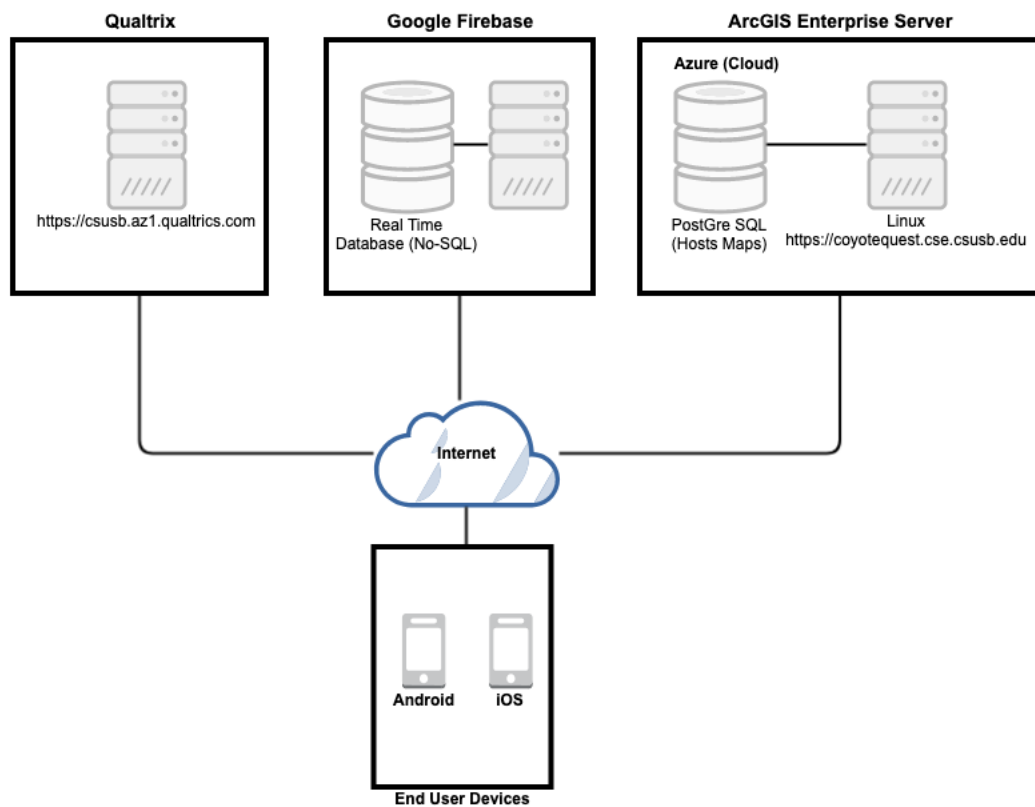
Name	Role	Responsibilities
Ginger Hartman	Sponsor	- Approve project scope, deliverables, budget
Dr. Conrad Shayo	Project Advisor	-Approve project scope, deliverables, UML/UP process
Thomas Saldana	iOS Developer	-Gather requirements -Define Project Scope -Human Resource Management -Project Communication -Implement UML/UP process -iOS development -Project Manager
Mohamed Rahman	Android Developer	-Android developer
Eric Niles	ArcGIS Cartographer/Designer	-Map builder
Viridiana Guzman	Designer	-Application designer
Birdy Wang	IT/Cloud Support at CSUSB	-ArcGIS Enterprise Server Administrator
Danny Vasquez	IT Technical Consultant at CSUSB	-Qualtrics Administrator

Note: The above table shows the resource allocation in the scope management plan.

Appendix B



Note. This diagram shows an overview of the Ability Sports & Education Festival Backend Infrastructure.



Note. This diagram shows the Ability Sports & Education Festival Network Diagram

Appendix C

ID	Task Name	Duration	Start	Finish	Predecessors	Resource Names	% Complete
1	Gather Requirements	14 days	Mon 1/21/19	Sat 2/2/19		Thomas Saldana	100%
2	Design UML/UP Process	7 days	Sat 2/2/19	Fri 2/8/19	1	Thomas Saldana	100%
3	Analysis & Effort Estimation	7 days	Fri 2/8/19	Thu 2/14/19	2	Thomas Saldana	100%
4	Design and Implementation (iOS)	35 days	Thu 2/14/19	Sun 3/17/19	3		82%
5	Create Initial Project and Side Menu	14 days	Thu 2/14/19	Wed 2/27/19	3	Thomas Saldana	100%
6	Create Home Screen	3 days	Wed 2/27/19	Fri 3/1/19	5	Thomas Saldana	100%
7	Create New View/Class and Point to Registration Page	4 days	Sat 3/2/19	Tue 3/5/19	6	Thomas Saldana	100%
8	Create Additional Views and Classes	7 days	Tue 3/5/19	Mon 3/11/19	7	Thomas Saldana	100%
9	Testing	7 days	Mon 3/11/19	Sun 3/17/19	8	Thomas Saldana	10%
10	Design and Implementation (Android)	35 days	Thu 2/14/19	Sun 3/17/19	3		72%
11	Create Initial Project and Navigation Drawer	14 days	Thu 2/14/19	Wed 2/27/19	3	Mohamed Rahma	100%
12	Create Home Screen	3 days	Wed 2/27/19	Fri 3/1/19	5	Mohamed Rahma	100%
13	Point to Registration Page	4 days	Sat 3/2/19	Tue 3/5/19	6	Mohamed Rahma	10%
14	Create Additional Activities and Classes	7 days	Tue 3/5/19	Mon 3/11/19	7	Mohamed Rahma	100%
15	Testing	7 days	Mon 3/11/19	Sun 3/17/19	8	Mohamed Rahma	10%

Table 1.2, iOS Development Schedule Winter 2019

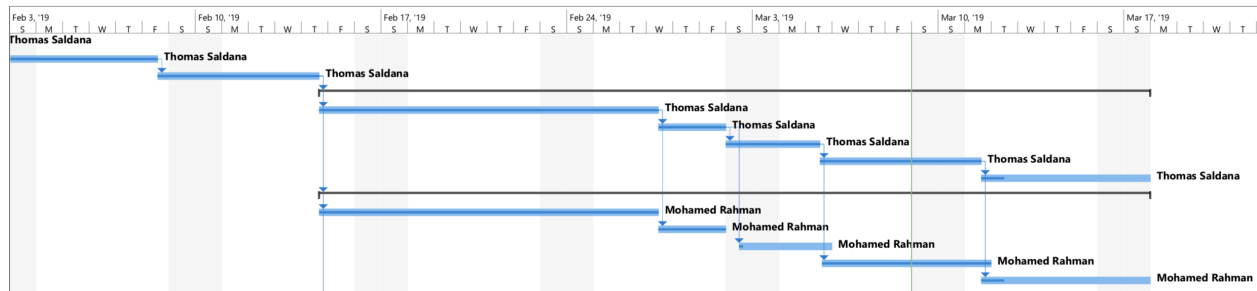


Table 1.4, Gantt Chart of Winter 2019 Schedule

Appendix D

Functional Requirements

ID	Details	Type	Priority
R1	The Ability Festival system shall register participants during the event.	Functional User Account	Must Have
R2	The Ability Festival system shall allow users to logon.	Functional User Account	Must Have
R3	The Ability Festival system shall allow users to logoff.	Functional User Account	Must Have
R4	<p>The Ability Festival system shall collect the following information upon registration:</p> <ul style="list-style-type: none"> • *First Name: • *Last Name: • *Email Address: • *Address 1: • *City: • *ZIP Code: • *State: • Gender • Age • Phone: • The phone number is: • Shirt Size • Allergies 	Functional User Account	Must Have
R5	The Ability Festival system shall ensure account passwords meet the following complexity: minimum 10 characters, with at least one upper case, lowercase, number and special character.	Non-Functional User Account	Must Have
R6	The Ability Festival system shall allow users to reset their passwords	Functional User Account	Must Have
R7	The Ability Festival system shall allow for the deletion of an account.	Functional User Account	Should-Have

R8	The Ability Festival system shall allow users to log on to their account.	Functional User Account	Must Have
R9	The Ability Festival system shall remain operational 1 week before and after the event	Non-Functional Availability	Must-Have
R10	The Ability Festival system shall be accessible via any device with internet access.	Non-Functional Availability	Should-Have
R11	The Ability Festival system shall use Azure to host the ArcGIS server and ArcGIS maps	Non-Functional Infrastructure	Must Have
R12	The Ability Festival system shall allow the client to modify certain entries such as events and coaches	Functional Firebase Database Entry	Must Have
R13	The Ability Festival system shall display information on events, coaches, maps, sponsors, and the event	Functional Usability	Must Have
R14	The Ability Festival system shall provide search results within 3 seconds	Non-Functional Usability	Must Have
R15	The Ability Festival system shall be accessible by all users with disabilities	Functional Usability	Must Have
R16	The Ability Festival system shall provide an interface that is both appealing and easy to use	Non-Functional Usability	Must Have
R17	The Ability Festival system shall provide a responsive interface.	Non-Functional Usability	Must Have
R18	The Ability Festival system shall perform analytics on the raw data using FireBase Analytics	Functional Analytics	Want to Have
R19	The Ability Festival system shall display events within a geographical area using ArcGIS	Non-Functional Analytics	Must Have

R20	The Ability Festival system shall be able to keep track of users who register for the event	Non-Functional Analytics	Must Have
R21	The Ability Festival system shall use Google's FireBase as its back end real time database	Functional Database Entry	Must Have
R22	The Ability Festival system shall use Qualitrix as its backend registration system	Functional Database Entry	Must Have
R23	The Ability Festival system shall include a Map of the campus with pins marking the location of all clinics inside buildings and outside	Functional ArcGIS	Must Have
R24	The Ability Festival system shall have guide users when they click a location in the ArcGIS map	Functional ArcGIS	Should Have
R25	The Ability Festival system shall be created for both the Android and iOS platforms	Functional ArcGIS	Must Have
R26	The Ability Festival system shall hold the coach's bios and pictures that corresponds to each sport. (accessible by voice and text)	Functional ArcGIS	Must Have
R27	The Ability Festival system shall be able to create an adapted evacuation plan and store it in the app with the option to send someone who may provide assistance	Functional ArcGIS	Should Have
R28	The Ability Festival system shall specifically label which bathrooms can accommodate adult changing areas	Functional ArcGIS	Must Have
R29	The Ability Festival system shall Map location of all bathrooms available	Functional ArcGIS	Must Have
R30	The Ability Festival system shall display the location of quiet rooms using ArcGIS	Functional ArcGIS	Should Have
R31	The Ability Festival system shall Map the location of water stations	Functional ArcGIS	Must Have
R32	The Ability Festival system shall Map the location of First Aid tents	Functional ArcGIS	Must Have

R33	The Ability Festival system shall Houses FAQ's and link to website/registration and access account	Functional ArcGIS	Should Have
R34	The Ability Festival system shall display all sports in the event	Functional	Must Have
R35	The Ability Festival system shall display all coaches who will coach during the event	Functional	Must Have
R36	The Ability Festival system shall provide a page for the key-note speaker	Functional	Must Have
R37	The Ability Festival system shall display a resources and other services page	Functional	Should Have
R37	The Ability Festival system shall have a side menu that appears in every view of the application	Functional	Must Have
R38	The Ability Festival system shall have a custom tool bar	Functional	Want to Have
R39	The Ability Festival system shall have a home button I every view of the app	Functional	Should Have
R40	The Ability Festival system shall be developed for both the iOS and Android operating systems	Functional	Must Have

Appendix E

Actors

<i>Actor</i>	<i>Semantics</i>
<i>Event User</i>	<p>User who will primarily be using the application on the day of the event</p> <p>Synonyms: User, participant, volunteer</p> <p>Homonyms: None</p>
<i>Casual User</i>	<p>Users who will use the application casually all year long, specifically for event information</p> <p>Synonyms: User, participant</p> <p>Homonyms: None</p>
<i>Project Manager</i>	<p>A person who over sees the overall technical administration of the Ability Sports Festival application</p> <p>Synonyms: Employee, user, manager</p> <p>Homonyms: None</p>
<i>iOS Developer</i>	<p>A person who develops the Ability Sports & Education Festival application for the iOS operating system</p> <p>Synonyms: Coder, developer,</p> <p>Homonyms: None</p>
<i>Android Developer</i>	<p>A person who develops the Ability Sports & Education Festival application for the Android operating system</p> <p>Synonyms: Coder, developer,</p> <p>Homonyms: None</p>
<i>Volunteer</i>	<p>A user who is a volunteer and uses the app on the day of the event</p> <p>Synonyms: None</p>

	Homonyms: None
<i>Cartographer</i>	<p>A person who builds ArcGIS maps using ArcMap, ArcGIS Online, and ArcGIS Pro</p> <p>Synonyms: None</p> <p>Homonyms: None</p>
<i>ArcGIS Server Administrator</i>	<p>A person who administers the ArcGIS server</p> <p>Synonyms: None</p> <p>Homonyms: None</p>
<i>Systems Administrator</i>	<p>A person who administers the overall Ability Festival application, both front-end and back-end</p> <p>Synonyms: None</p> <p>Homonyms: None</p>
<i>Qualtrics Administrator</i>	<p>A technical person who administers the Qualtrics production servers</p> <p>Synonyms: System Administrator, DBA</p> <p>Homonyms: None</p>
<i>Cloud Support Engineer</i>	<p>A person who administers the ArcGIS Enterprise server on Azure</p> <p>Synonyms: System Administrator, IT Technical Consultant, Systems Engineer</p> <p>Homonyms: None</p>

Appendix F

Glossary

<i>Term</i>	<i>Definition</i>
<i>Account</i>	<p>An interface where users can add and manage their data such as usernames and passwords. This data is stored in the Qualitrix back end database</p> <p>Synonyms: None</p> <p>Homonyms: None</p>
<i>ArcGIS Map</i>	<p>The primary map that will show all locations in the event</p> <p>Synonyms: None</p> <p>Homonyms: None</p>
<i>Events</i>	<p>The main view that users will use to find events during the Ability Festival</p> <p>Synonyms: Bill</p> <p>Homonyms: None</p>
<i>Log On</i>	<p>The act of authenticating with the Qualitrix system and creating a session.</p> <p>Synonyms: Log In, Sign In</p> <p>Homonyms: None</p>
<i>Log Off</i>	<p>The act of terminating a session with Qualitrix system.</p> <p>Synonyms: Log Out, Sign Out</p> <p>Homonyms: None</p>
<i>Side View</i>	<p>A view that slides from left side of any screen in the app</p> <p>Synonyms: None</p> <p>Homonyms: None</p>
<i>FireBase</i>	<p>A backend, cloud based, Mobile development service</p>

	Synonyms: None
	Homonyms: None
<i>Real Time Database</i>	A real time database hosted on firebase that will host application information
	Synonyms: None
	Homonyms: None
<i>ArcGIS</i>	A suit of geographic information systems
	Synonyms: None
	Homonyms: None
<i>iOS</i>	Apple iPhone operating system
	Synonyms: Operating System, OS
	Homonyms: None
<i>Android</i>	Android operating system
	Synonyms: Android, phone, operating system
	Homonyms: None
<i>ArcMap</i>	An ArcGIS application that allows cartographers to build maps
	Synonyms: None
	Homonyms: None
<i>ArcGIS Pro</i>	An ArcGIS web-based application that allows cartographers to build maps
	Synonyms: None
	Homonyms: None
<i>ArcGIS Enterprise</i>	A suit of ArcGIS server software used to host maps and other ArcGIS services

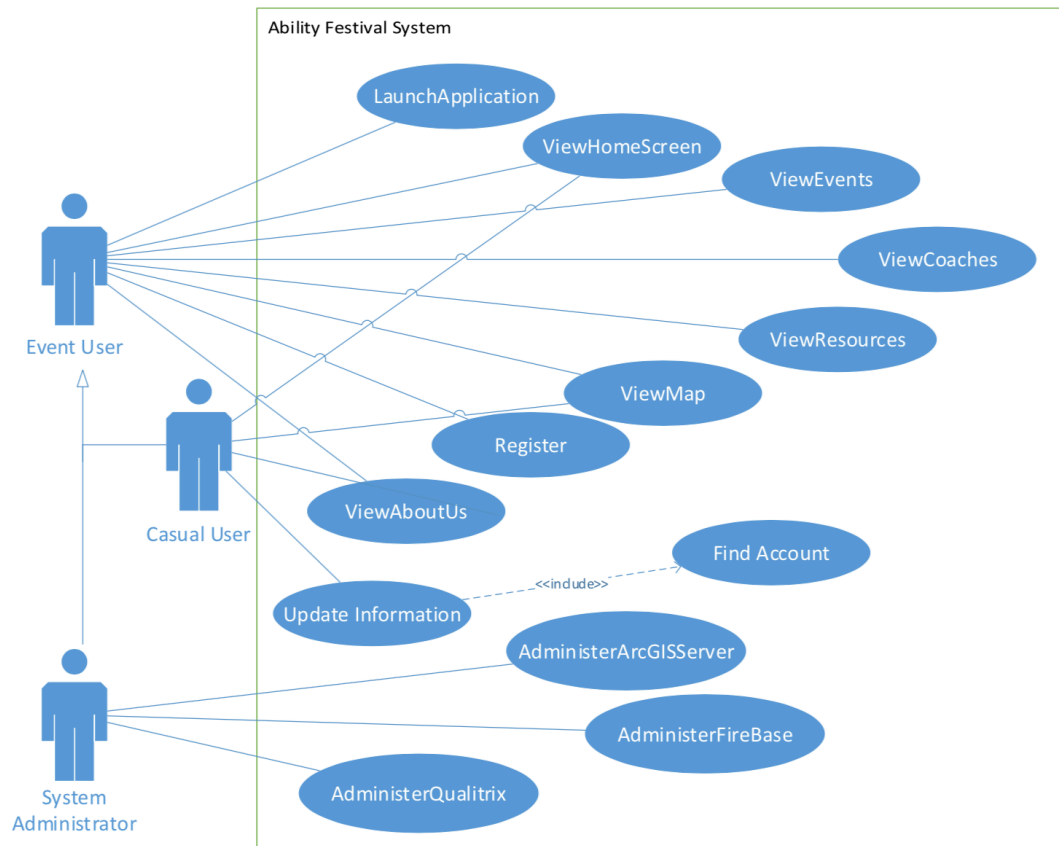
	<p>Synonyms: ArcGIS Server, Server</p> <p>Homonyms: None</p>
<i>Geodatabase</i>	<p>A special type of database that holds all types of geospatial data such as feature classes and data sets</p> <p>Synonyms: Gatabase, personal geodatabase, file geodatabase</p> <p>Homonyms: None</p>
<i>Network Analyst</i>	<p>An ArcMap and ArcGIS Pro extension that allows you to build a network datasets and perform analysis on network data sets</p> <p>Synonyms: routing extension</p> <p>Homonyms: None</p>
<i>Address Locator</i>	<p>An ArcMap and ArcGIS Pro extension that allows you to build a network datasets and perform analysis on network data sets</p> <p>Synonyms: locator</p> <p>Homonyms: None</p>
<i>Geocoding</i>	<p>The process of taking a location on the surface of the earth and assigning it a human readable address</p> <p>Synonyms: None</p> <p>Homonyms: None</p>

Appendix G

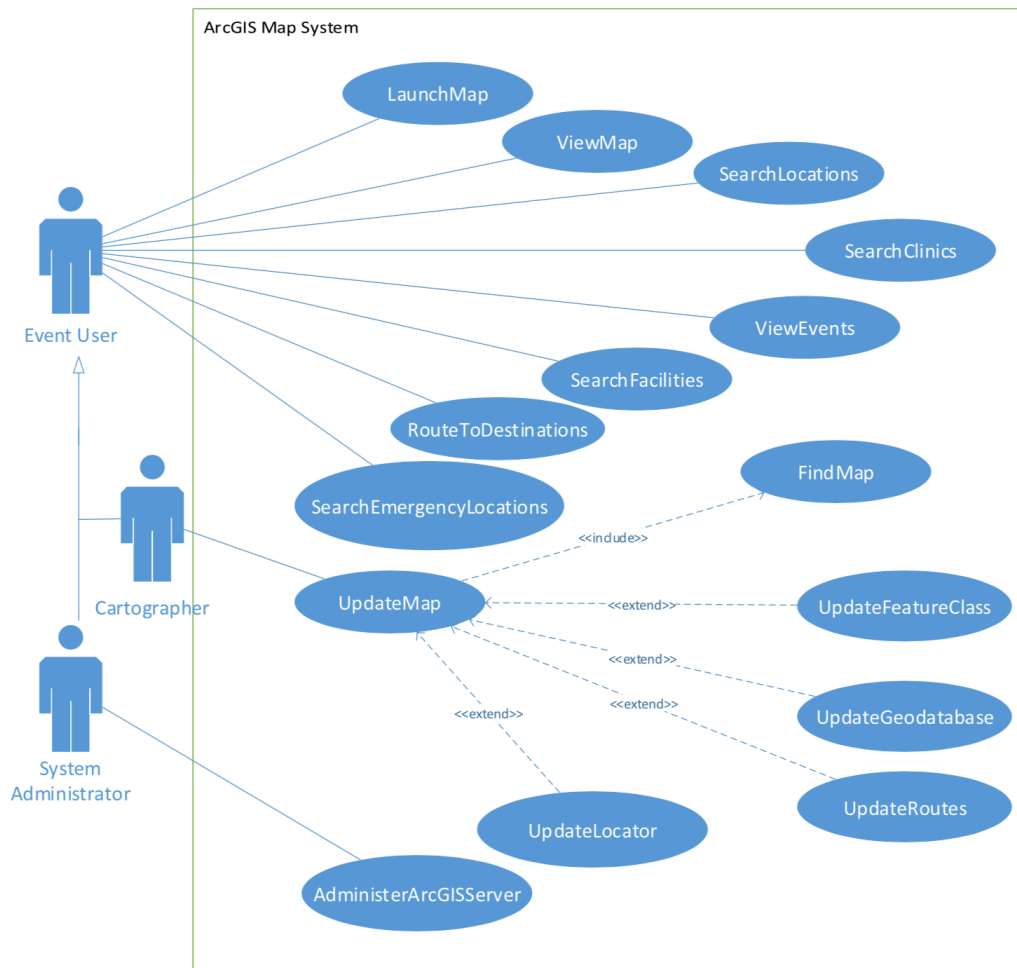
Use Cases

<i>Use Case</i>	<i>Brief Semantics</i>
<i>Log On</i>	All users will log on and authenticate against Firebase
<i>Log off</i>	All users will log of or will be allowed to stay logged on if they choose
<i>Register</i>	Users can register for the event using the Register Now button at the bottom of the home page
<i>DeleteAccount</i>	A user can elect to delete his or her account.
<i>FindAccount</i>	Users can use the app without an account. Accounts will be found upon user request
<i>Search Event</i>	Users can search for events using the Events button
<i>Search Coaches</i>	Users can search for coaches using the Coaches button
<i>Search Sports Clinics</i>	Users can search for sports clinics using the Sports Clinics button and ArcGIS map
<i>Search Resources</i>	Users can search event resources using the Resources button
<i>Search About Us</i>	Users can search more information about the even using the About Us button
<i>Search Map</i>	Users can search all sort of information using the apps ArcGIS map

Appendix H



Note. This Use Case diagrams shows the Ability Sports & Education Festival System.



Note. This Use Case diagrams shows the ArcGIS Map System.

Appendix I

Use Case Specifications

Use Case: LogOnUser
Use Case ID: 1
Brief Description: The User can authenticate against Firebase. All users will see the same information when logged in.
Primary Actors: Event Users, Casual Users
Secondary Actors: None
Preconditions: 1. The User is not logged on into the system.
Main Flow: 1. The User selects “Log on”. 2. The User enters their username and password. 3. While the User credentials are invalid 3.1 The User is required to re-enter the username and password. 3.2 The User credentials are validated 4. The User is logged onto the system
Postconditions: 1. The User is logged into the system and their account.

Alternative flows: InvalidUserName InvalidPassword Cancel

Alternative flow: LogOnUser:InvalidUserName
Use Case ID: 1.1
Brief Description: The User has entered an invalid username to login and has been notified.
Primary Actors: Event Users, Casual Users
Secondary Actors: None
Preconditions: 1. The User has submitted an invalid username.
Main Flow: 1. The User receives a message that they have entered an invalid username.
Postconditions: None

Alternative flow: LogOnUser:InvalidPassword
Use Case ID: 1.2
Brief Description: The User has entered an invalid password to login and is notified.
Primary Actors: Event Users, Casual Users
Secondary Actors: None
Preconditions: 1. The User has submitted an invalid password.
Main Flow: 1. The User receives a message that they have entered an invalid password.
Postconditions: None

Alternative flow: LogOnUser:Cancel
Use Case ID: 1.3
Brief Description: <p>The User cancels the logon process.</p>
Primary Actors: <p>Event Users, Casual Users</p>
Secondary Actors: <p>None</p>
Preconditions: <p>None</p>
Main Flow: <ol style="list-style-type: none"> 1. The alternative flow begins when the user initiates the cancellation. 2. The User cancels the logon process.
Postconditions: <ol style="list-style-type: none"> 1. The user does not get logged on to the system and their account.

Use Case: LogOffUser
Use Case ID: 2
Brief Description: A user can terminate their current session with the system.
Primary Actors: Event Users, Casual Users
Secondary Actors: None
Preconditions: 1. The user is logged into the system.
Main Flow: 1. The user selects the “Log off” button on the top-right of the application. 2. The user is logged off the system.
Postconditions: None

Use Case: Register User
Use Case ID: 3
Brief Description: The User can register for the event
Primary Actors: Event Users
Secondary Actors: None
Preconditions: 1. The User has launched the app
Main Flow: 1. The Users selects “Register Now” at the bottom of the main home page 2. The User enters his or her information on the Qualitrix web page 3. The User chooses what information to register 4. The User information is updated.
Postconditions: 1. The User information is updated.
Alternative flows: None

Use Case: DeleteAccount
Use Case ID: 4
Brief Description: The User deletes their active account.
Primary Actors: Event Users, Casual Users
Secondary Actors: None
Preconditions: 1. The User is logged into the application
Main Flow: 1. include(FindAccount) 2. The User selects “delete account”. 3. If the User selects “yes” to confirm deletion, then 3.1. The User deletes their account 4. If the User select “no” to deleting their account, then 4.1. The User account is not deleted
Postconditions: 1. The User account is deleted from the system
Alternative flows: None

Use Case: FindAccount
Use Case ID: 6
Brief Description: The system locates the user account.
Primary Actors: Event Users, Casual Users
Secondary Actors: None
Preconditions: 1. The user is logged into the system.
Main Flow: 1. This use case starts when a User selects “Update Account” or “Delete Account”. 2. If the user is a registered user 2.1. The system requests the account credentials. 2.2. The system locates the account related to the account number. 3. Else 3.1. The system locates the account based on the users that is currently logged in.
Postconditions: 1. The system has found the user account.

Use Case: Search Event
Use Case ID: 7
Brief Description: The Event user will press on the <i>Events</i> button on the main home page and he or she will be navigated to the Events view where he or she can view all the events.
Primary Actors: Event Users
Secondary Actors: None
Preconditions: 1. The user has launched the app.
Main Flow: 1. This use case starts when a User presses the <i>Events</i> button. 2. If the user is an Event User, the user is registered 2.1. The system locates and pulls all event information from the Firebase real time database 3. Else 3.1. The system displays static information
Postconditions: None

Use Case: SearchCoaches
Use Case ID: 8
Brief Description: The Event User will press on the <i>Coaches</i> button on the main home page and he or she will be navigated to the coaches view where he or she can view all the coaches
Primary Actors: Event Users
Secondary Actors: None
Preconditions: 1. The user is logged into the system.
Main Flow: 1. This use case starts when a User presses the “ <i>Coaches</i> ” button. 2. If the user is an Event User, the user is registered 2.1. The system locates and pulls all coaches information from the Firebase real time database 3. Else 3.1. The system displays static information from the phone
Postconditions: None

Extension Use Case: SearchSportsClinics
Use Case ID: 9
Brief Description: The Event User will press on the <i>Sports Clinics</i> button on the main home page and he or she will be navigated to the coaches view where he or she can view all the sports clinics'
Primary Actors: Event Users
Secondary Actors: None
Preconditions: The user has launched the application
Main Flow: <ol style="list-style-type: none"> 1. This use case starts when a User presses the Sports Clinics button. 2. If the user is an Event User, the user is registered <ol style="list-style-type: none"> 2.1. The system locates and pulls all clinic's information from the Firebase real time database 3. Else <ol style="list-style-type: none"> 3.1. The system displays static information from the phone
Postconditions: None

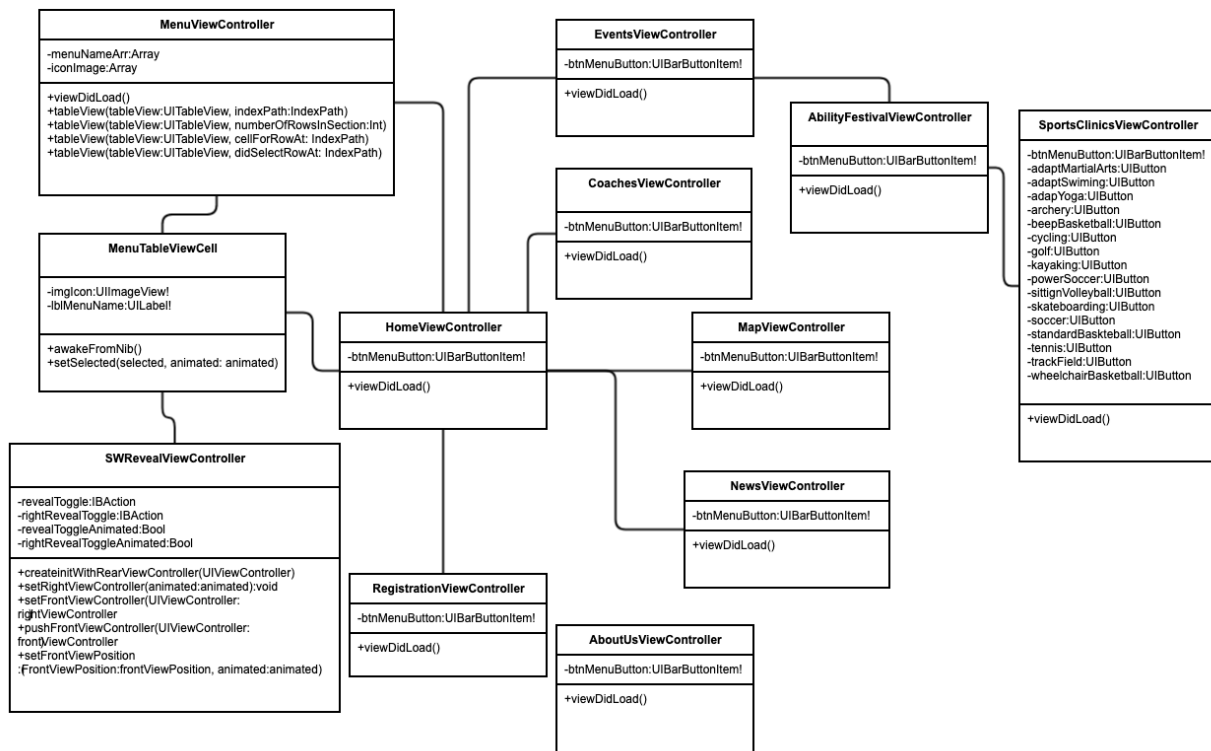
Use Case: Search Resources
Use Case ID: 10
Brief Description: The Event User will be able to search any information or resources from the side menu
Primary Actors: Event Users
Secondary Actors: None
Preconditions: 1. The user slides the side menu
Main Flow: 1. The User finds a link on the side menu 2. The User clicks on the link and is navigated to a new view
Postconditions: None

Extension Use Case: Search About Us
Use Case ID: 11
Brief Description: Segment 1: The results are aggregated and analyzed.
Primary Actors: Event User
Secondary Actors: None
Segment 1 Preconditions: 1. Information about the Event is requested from FireBase
Segment 1 Flow: 1. The results are processed and displayed on the About US view 2. Users can click link in the About Us view that will show them more information about the event
Segment 1 Postconditions: 1. The analyzed results are returned

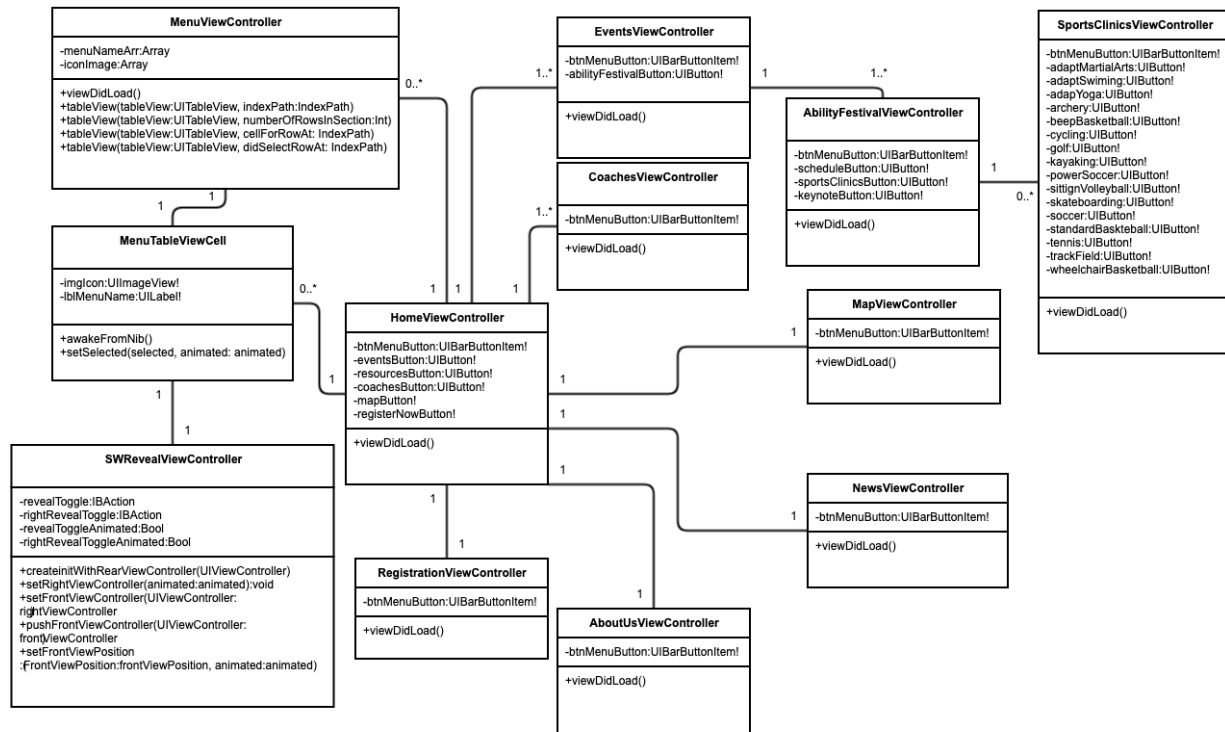
Use Case: Search Map
Use Case ID: 12
Brief Description: Users will launch the ArcGIS map and be able search event locations
Primary Actors: ArcGIS User
Secondary Actors: None
Preconditions: 1. The ArcGIS map has been launched
Main Flow: 1. The user sees the map of the event 2. For each ArcGIS user 2.1. Search event locations Extension point: FilterSearchResults Extension point: AnalyzeData 2.2. Information is Displayed
Postconditions: 1. A new query is made

Appendix J

Preliminary Class Diagram



Extended Class Diagram



Appendix K

Code

MenuViewController.swift

```
//  
// MenuViewController.swift  
// SideMenu3  
//  
// Created by thomassldn on 2/26/19.  
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.  
//  
  
import UIKit  
  
class MenuViewController: UIViewController, UITableViewDelegate, UITableViewDataSource  
{  
  
    //@IBOutlet weak var imgProfile: UIImageView!  
  
    var menuNameArr: Array = [String]()  
    var iconImage:Array  = [UIImage]()  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
  
        menuNameArr = [ "Home", "Events", "Coaches", "News", "Map", "About Us"]  
        iconImage = [UIImage(named: "Home")!,UIImage(named: "events")!,UIImage(named:  
"coaches")!,UIImage(named: "events")!,UIImage(named: "map")!,UIImage(named: "coaches")!]  
  
        //The following code sets up the user profile picture in the side menu  
  
        imgProfile.layer.borderColor = UIColor.blue.cgColor  
        imgProfile.layer.borderWidth = 2  
        //imgProfile.layer.cornerRadius = 60  
        imgProfile.layer.masksToBounds = false  
        imgProfile.clipsToBounds = true  
  
    }  
  
    func tableView(_ tableView: UITableView, numberOfRowsInSectionSection section: Int) -> Int {
```

```

        return menuNameArr.count
    }

    func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
    UITableViewCell {
        let cell = tableView.dequeueReusableCell(withIdentifier: "MenuTableViewCell") as!
    MenuTableViewCell

        cell.imgIcon.image = iconImage[indexPath.row]
        cell.lblMenuName.text! = menuNameArr[indexPath.row]

        return cell
    }

    func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {

        let revealViewController:SWRevealViewController = self.revealViewController()

        let cell:MenuTableViewCell = tableView.cellForRow(at: indexPath) as!
    MenuTableViewCell

        //HOME
        if cell.lblMenuName.text! == "Home" {

            let mainStoryboard:UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
            let desController = mainStoryboard.instantiateViewController(withIdentifier:
"ViewController") as! ViewController

            let newFrontViewController =
    UINavigationController.init(rootViewController:desController)

            revealViewController.pushFrontViewController(newFrontViewController, animated:
true)

        }

        //EVENTS
        if cell.lblMenuName.text! == "Events" {

            let mainStoryboard:UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
            let desController = mainStoryboard.instantiateViewController(withIdentifier:
"EventsViewController") as! EventsViewController

            let newFrontViewController =
    UINavigationController.init(rootViewController:desController)

```

```

        revealViewController.pushFrontViewController(newFrontViewController, animated:
true)
    }

    //COACHES
    if cell.lblMenuName.text! == "Coaches" {

        let mainStoryboard: UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
        let desController = mainStoryboard.instantiateViewController(withIdentifier:
"CoachesViewController") as! CoachesViewController

        let newFrontViewController =
UINavigationController.init(rootViewController: desController)

        revealViewController.pushFrontViewController(newFrontViewController, animated:
true)
    }

    //NEWS
    if cell.lblMenuName.text! == "News" {

        let mainStoryboard: UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
        let desController = mainStoryboard.instantiateViewController(withIdentifier:
"NewsViewController") as! NewsViewController

        let newFrontViewController =
UINavigationController.init(rootViewController: desController)

        revealViewController.pushFrontViewController(newFrontViewController, animated:
true)
    }

    //MAP
    if cell.lblMenuName.text! == "Map" {

        let mainStoryboard: UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
        let desController = mainStoryboard.instantiateViewController(withIdentifier:
"MapViewController") as! MapViewController

        let newFrontViewController =
UINavigationController.init(rootViewController: desController)

```



```

        revealViewController.pushFrontViewController(newFrontViewController, animated:
true)
    }

    //ABOUT US
    if cell.lblMenuName.text! == "About Us" {

        let mainStoryboard: UIStoryboard = UIStoryboard(name: "Main", bundle: nil)
        let desController = mainStoryboard.instantiateViewController(withIdentifier:
"AboutUsViewController") as! AboutUsViewController

        let newFrontViewController =
UINavigationController.init(rootViewController:desController)

        revealViewController.pushFrontViewController(newFrontViewController, animated:
true)
    }

}
}
}

```

MenuTableViewCell.swift

```

//
// MenuTableViewCell.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/2/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

```

```

import UIKit

```

```

class MenuTableViewCell: UITableViewCell {

    @IBOutlet weak var imgIcon: UIImageView!
    @IBOutlet weak var lblMenuName: UILabel!

    override func awakeFromNib() {
        super.awakeFromNib()
        // Initialization code
    }
}

```

```

    }

    override func setSelected(_ selected: Bool, animated: Bool) {
        super.setSelected(selected, animated: animated)

        // Configure the view for the selected state
    }
}

```

HomeController.swift

```

//
// ViewController.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/2/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

```

```
import UIKit
```

```
class ViewController: UIViewController {
```

```

    @IBOutlet weak var btnMenuButton: UIBarButtonItem!
    @IBOutlet weak var eventsButton: UIButton!
    @IBOutlet weak var coachesButton: UIButton!
    @IBOutlet weak var resourcesButton: UIButton!
    @IBOutlet weak var mapButton: UIButton!
    @IBOutlet weak var registerNowButton: UIButton!

```

```

    override func viewDidLoad() {
        super.viewDidLoad()

```

```

        btnMenuButton.target = revealViewController()
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))

```

```

        //make eventsButton round
        eventsButton.layer.cornerRadius = 10.0
        eventsButton.layer.masksToBounds = true

```

```

        //make coachesButton round
        coachesButton.layer.cornerRadius = 10.0
        coachesButton.layer.masksToBounds = true

```

```

//make resourcesButton round
resourcesButton.layer.cornerRadius = 10.0
resourcesButton.layer.masksToBounds = true

//make mapButton round
mapButton.layer.cornerRadius = 10.0
mapButton.layer.masksToBounds = true

//make registerNowButton round
registerNowButton.layer.cornerRadius = 10.0
registerNowButton.layer.masksToBounds = true

}

}

```

EventsViewController.swift

```

//
// EventsViewController.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/4/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

```

```
import UIKit
```

```
class EventsViewController: UIViewController {
```

```
    @IBOutlet weak var btnMenuButton: UIBarButtonItem!
```

```
    @IBOutlet weak var abilityFestivalButton: UIButton!
```

```
    override func viewDidLoad() {
```

```
        super.viewDidLoad()
```

```
        btnMenuButton.target = revealViewController()
```

```
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))
```

```
        //make abilityFestivalButton round
```

```
        abilityFestivalButton.layer.cornerRadius = 10.0
```

```
        abilityFestivalButton.layer.masksToBounds = true
    }
}

```

```

    }

    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before
    navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destination.
        // Pass the selected object to the new view controller.
    }
    */
}

```

CoachesViewController.swift

```

//
// CoachesViewController.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/4/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

```

import UIKit

```

class CoachesViewController: UIViewController {

    @IBOutlet weak var btnMenuButton: UIBarButtonItem!
    override func viewDidLoad() {
        super.viewDidLoad()
        btnMenuButton.target = revealViewController()
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))
    }

}

```

MapViewController.swift

```

//

```

```

// MapViewController.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/4/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

import UIKit
import ArcGIS

class MapViewController: UIViewController {

    //side menu button
    @IBOutlet weak var btnMenuButton: UIBarButtonItem!

    //Map view outlet
    @IBOutlet weak var mapView: AGSMapView!
    //Used to set intial map view
    var map:AGSMap!

    override func viewDidLoad() {
        super.viewDidLoad()

        //Display a map using the ArcGIS Online imagery basemap service
        //The map originally displays the csusb campus, coordinates latitude: 34.181921, longitude:
-117.320019
        self.mapView.map = AGSMap(basemapType: .streets, latitude: 34.181921, longitude: -
117.320019, levelOfDetail: 17)

        btnMenuButton.target = revealViewController()
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))

    }

}

```

NewsViewController.swift

```

//
// NewsViewController.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/4/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.

```

```
//
```

```
import UIKit
```

```
class NewsViewController: UIViewController {
```

```
    @IBOutlet weak var btnMenuButton: UIBarButtonItem!
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()
```

```
        btnMenuButton.target = revealViewController()  
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))  
    }
```

```
}
```

```
AboutUsViewController.swift
```

```
//
```

```
// AboutUsViewController.swift
```

```
// AbilitSports&EducationFestival
```

```
//
```

```
// Created by thomassldn on 3/4/19.
```

```
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
```

```
//
```

```
import UIKit
```

```
class AboutUsViewController: UIViewController {
```

```
    @IBOutlet weak var btnMenuButton: UIBarButtonItem!
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()
```

```
        btnMenuButton.target = revealViewController()  
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))  
    }
```

```
}
```

```
AbilityFestivalViewController.swift
```

```
//
```

```
// SchedulesViewController.swift
```

```
// AbilitSports&EducationFestival
```

```

//
// Created by thomassldn on 3/10/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

import UIKit

class AbilityFestivalViewController: UIViewController {

    @IBOutlet weak var btnMenuButton: UIBarButtonItem!
    @IBOutlet weak var scheduleButton: UIButton!
    @IBOutlet weak var sportsClinicsButtons: UIButton!
    @IBOutlet weak var keynoteButton: UIButton!

    override func viewDidLoad() {
        super.viewDidLoad()

        btnMenuButton.target = revealViewController()
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))

        //make scheduleButton round
        scheduleButton.layer.cornerRadius = 10.0
        scheduleButton.layer.masksToBounds = true

        //make sportsClinicsButton round
        sportsClinicsButtons.layer.cornerRadius = 10.0
        sportsClinicsButtons.layer.masksToBounds = true

        //make keynoteButton round
        keynoteButton.layer.cornerRadius = 10.00
        keynoteButton.layer.masksToBounds = true
    }

    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before
    navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destination.
        // Pass the selected object to the new view controller.
    }
    */

```

```
}
```

SportsViewController.swift

```
//  
// SportsViewController.swift  
// AbilitSports&EducationFestival  
//  
// Created by Admin on 3/7/19.  
// Copyright © 2019 csusb. All rights reserved.  
//
```

```
import UIKit
```

```
class SportsViewController: UIViewController {
```

```
    @IBOutlet weak var btnMenuButton: UIBarButtonItem!  
    @IBOutlet weak var martialArtsButton: UIButton!  
    @IBOutlet weak var swimmingButton: UIButton!  
    @IBOutlet weak var wheelchairBasketballButton: UIButton!  
    @IBOutlet weak var yogaButton: UIButton!  
    @IBOutlet weak var archeryButton: UIButton!  
    @IBOutlet weak var beepBasketballButton: UIButton!  
    @IBOutlet weak var cyclingButton: UIButton!  
    @IBOutlet weak var golfButton: UIButton!  
    @IBOutlet weak var kayakingButton: UIButton!  
    @IBOutlet weak var powerSoccerButton: UIButton!  
    @IBOutlet weak var sittingVolleyballButton: UIButton!  
    @IBOutlet weak var skateboardingButton: UIButton!  
    @IBOutlet weak var soccerButton: UIButton!  
    @IBOutlet weak var standardBasketballButton: UIButton!  
    @IBOutlet weak var tennisButton: UIButton!  
    @IBOutlet weak var trackFieldButton: UIButton!  
    @IBOutlet weak var wallClimbingButton: UIButton!
```

```
    override func viewDidLoad() {  
        super.viewDidLoad()
```

```
        btnMenuButton.target = revealViewController()  
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))
```


martialArtsButton.layer.cornerRadius = 10.0
martialArtsButton.layer.masksToBounds = **true**

swimmingButton.layer.cornerRadius = 10.0
swimmingButton.layer.masksToBounds = **true**

wheelchairBasketballButton.layer.cornerRadius = 10.0
wheelchairBasketballButton.layer.masksToBounds = **true**

yogaButton.layer.cornerRadius = 10.0
yogaButton.layer.masksToBounds = **true**

archeryButton.layer.cornerRadius = 10.0
archeryButton.layer.masksToBounds = **true**

beepBasketballButton.layer.cornerRadius = 10.0
beepBasketballButton.layer.masksToBounds = **true**

cyclingButton.layer.cornerRadius = 10.0
cyclingButton.layer.masksToBounds = **true**

golfButton.layer.cornerRadius = 10.0
golfButton.layer.masksToBounds = **true**

kayakingButton.layer.cornerRadius = 10.0
kayakingButton.layer.masksToBounds = **true**

powerSoccerButton.layer.cornerRadius = 10.0
powerSoccerButton.layer.masksToBounds = **true**

sittingVolleyballButton.layer.cornerRadius = 10.0
sittingVolleyballButton.layer.masksToBounds = **true**

skateboardingButton.layer.cornerRadius = 10.0
skateboardingButton.layer.masksToBounds = **true**

soccerButton.layer.cornerRadius = 10.0
soccerButton.layer.masksToBounds = **true**

standardBasketballButton.layer.cornerRadius = 10.0
standardBasketballButton.layer.masksToBounds = **true**

tennisButton.layer.cornerRadius = 10.0
tennisButton.layer.masksToBounds = **true**

```

        trackFieldButton.layer.cornerRadius = 10.0
        trackFieldButton.layer.masksToBounds = true

        wallClimbingButton.layer.cornerRadius = 10.0
        wallClimbingButton.layer.masksToBounds = true

    }

    /*
    // MARK: - Navigation

    // In a storyboard-based application, you will often want to do a little preparation before
    navigation
    override func prepare(for segue: UIStoryboardSegue, sender: Any?) {
        // Get the new view controller using segue.destination.
        // Pass the selected object to the new view controller.
    }
    */
}

```

RegistrationViewController.swift

```

//
// RegistrationViewController.swift
// AbilitSports&EducationFestival
//
// Created by thomassldn on 3/7/19.
// Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//

import UIKit
import WebKit

class RegistrationViewController: UIViewController {

    @IBOutlet weak var WebView: WKWebView!

    override func viewDidLoad() {
        super.viewDidLoad()

        //https://csusb.az1.qualtrics.com/jfe/form/SV_e3EJNoDZ1eovQpv?Q_JFE=qdg
        let csusb_url =
        URL(string:"https://csusb.az1.qualtrics.com/jfe/form/SV_d530KmZzbuMkgM5")
    }
}

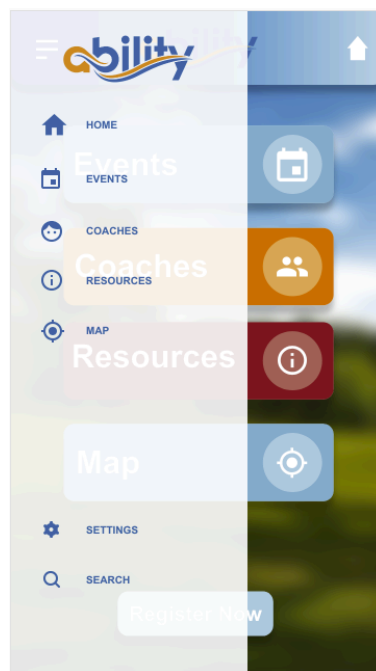
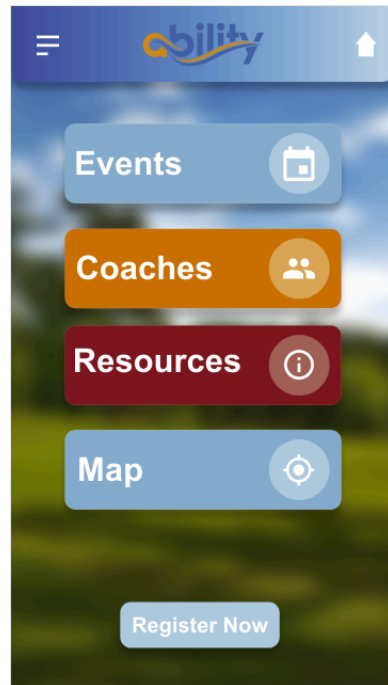
```

```
    let request = URLRequest(url: csusb_url!)
    WebView.load(request)
}
}
```

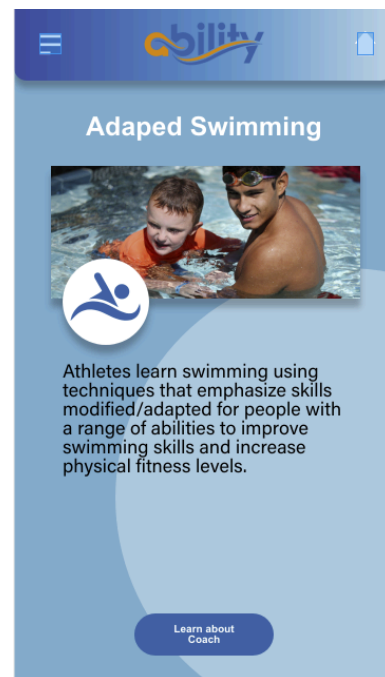
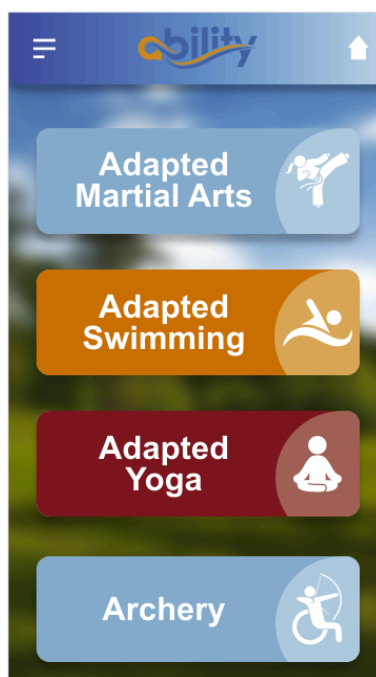
Appendix L


Application Design


Launch Screen & Home Screen




Ability Festival Events & Sports Mobile Application







Adaped Yoga




Teaching breathing, meditation and physical postures for people with a range of abilities to help relieve body pain, increase flexibility, improve strength and increase mental and emotional stability





Aaron "Wheelz" Fotheringham is a 26 year-old wheelchair moto-cross athlete from Las Vegas, Nevada. Aaron was born with Spina Bifida, which resulted in him having no usage of his legs. Aaron never let anything stop him. Even as a baby and small child, he did anything anyone else his age could do; he just had to figure out how to make it work for him. Aaron started riding at skate parks at the age of 8.



Over the last 10 years Aaron has challenged himself to try progressively more difficult tricks; carving, grinding, power-sliding, hand planting, and spinning are just a few of his accomplishments. In 2005, he perfected a mid air 180-degree turn. Then on July 13th, 2006, he landed the first wheelchair back flip. Four years later, at a camp in Woodward, he landed the first ever double back flip (August 26, 2010). Since then, he has gone on to perform it live while touring with the Nitro Circus. As if this is not enough, on February 9th, 2011, he landed his very first front flip in New Zealand, and on August 25, 2012, he shocked Brazilians by jumping and successfully landing a 50-ft gap off of the Mega Ramp in his chair. This year he landed his first ever 180 degree backflip at Woodward West.

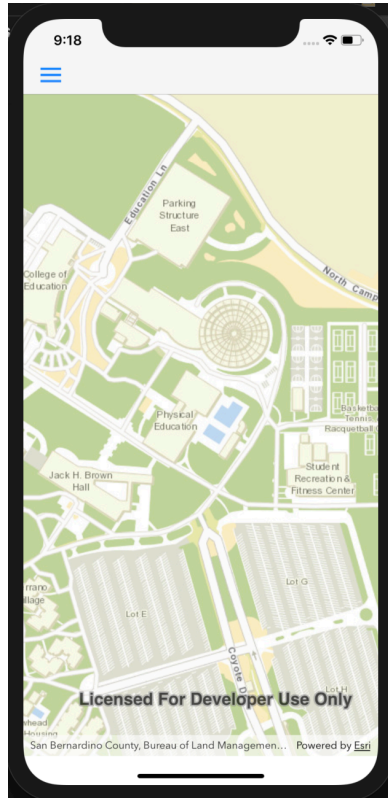
Aaron has a passion for what he does, and he wants to change the world's perception of people in wheelchairs, as well as help everyone see his/her own challenges in a new way. You certainly do not have to be disabled to be inspired by what he is able to do.





- Adapted Judo – PE 104
- Ambulatory Throwing – Track Field
- Archery – Upper Track Field
- Basketball (Standard) – Coussoulis Main Gym
- Basketball (Wheelchair) – Coussoulis Main Gym
- Beep Baseball/Softball – Upper Field
- Cycling & Hand cycling – Outdoor Basketball Courts
- Equine Therapy – Upper Baseball Field
- Floor Hockey - Rec Center (**Session 2 only**)
- Golf (ACS Golf Clinic) – Upper Field
- Human Foosball – Rec Center Tennis Courts
- Karate – HP room 104
- Kayaking - Pool (**Session 2 only**)
- Light Saber Academy – PE Building 100
- Motor Skills Tent – Rec Center Dance Studio 205
- Quad Rugby – Coussoulis Main Gym
- Rock Climbing – Vendor Village
- Rowing – Track Field
- Sitting Volleyball – Rec Center (**Session 1 only**)
- Skateboarding – Outdoor Basketball Courts
- Soccer – Lower Field
- Soccer Goal Kick by LA Galaxy Star Squad
- Street Hockey by the LA Kings Development Team
- Swimming – Pool (**Session 1 only**)
- Tennis – Tennis Courts
- Wheel Chair Motor Cross (WCMX) – Parking Lot G

Map & Registration Page



A screenshot of a mobile application showing the registration page for CSUSB. The page has a blue header with the CSUSB logo and a language dropdown menu set to 'English'. Below the header, there is a prompt: 'Please enter your first name and last name in the form below.' The form includes two input fields: 'First Name' with the value 'Thomas' and 'Last Name' with the value 'Saldana'. Below these fields, there is a question: 'What is your gender?' with two radio button options: 'Male' (selected) and 'Female'. Below this, there is another question: 'What is your year of birth?' with a text input field containing '1992'. At the bottom, there is a question: 'Are you Spanish, Hispanic, or Latino or none of these (select all that apply).' with two radio button options: 'Spanish' and 'Hispanic'. The status bar at the top shows the time as 9:19. At the bottom of the page, there are navigation arrows and a 'Done' button.

Appendix M

Create Initial Project

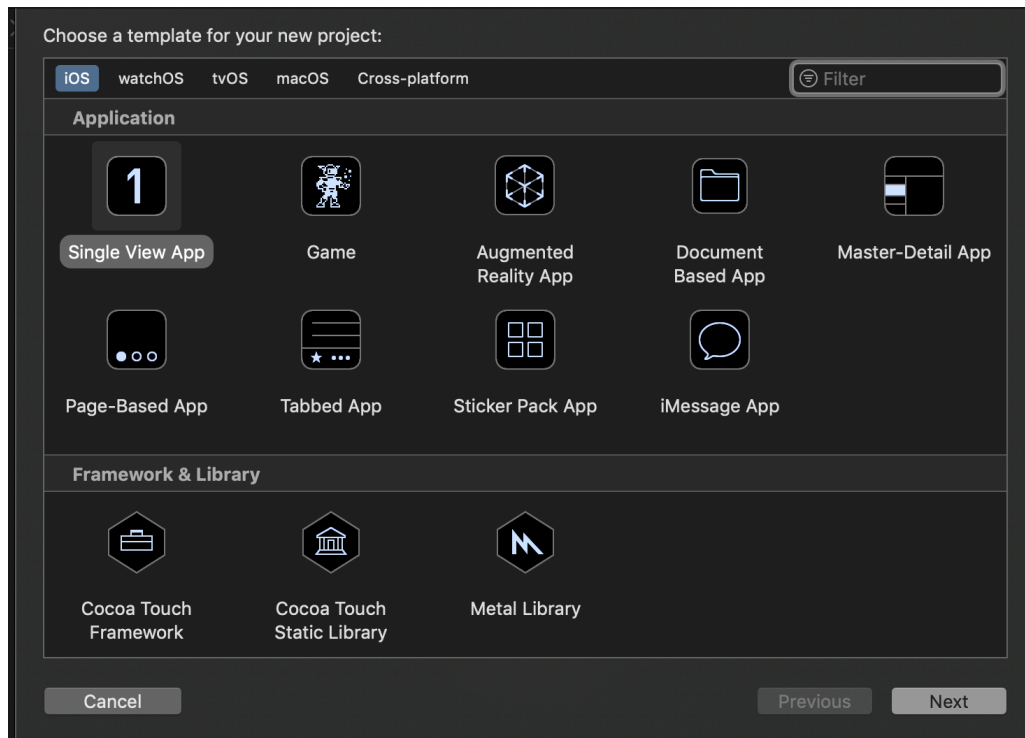
[1] Launch Xcode



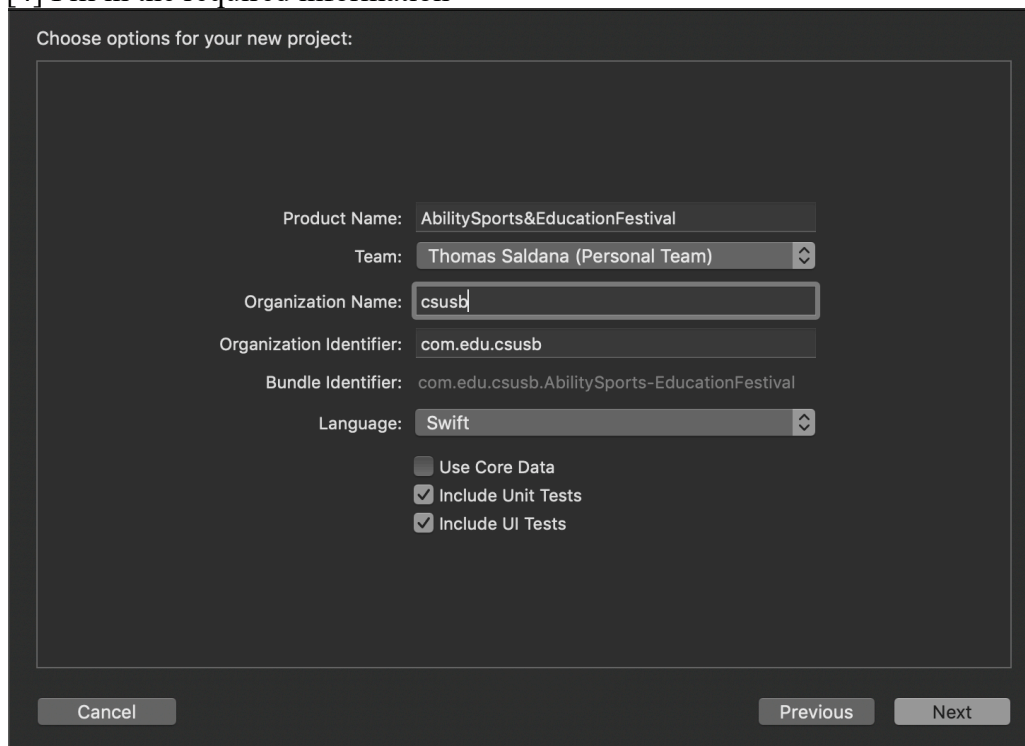
[2] Choose “Create a new Xcode project”



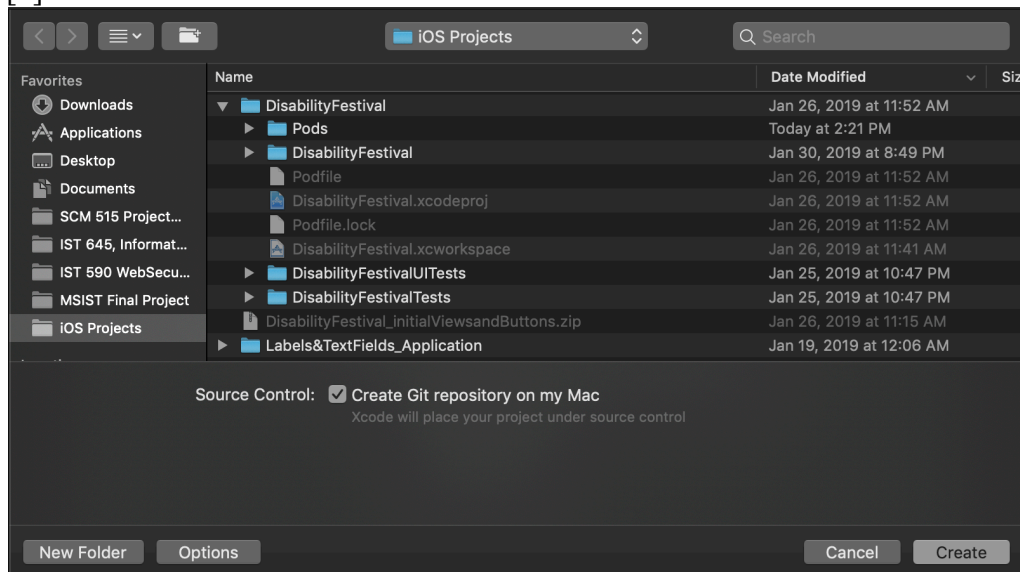
[3] Create a Single View application and click Next



[4] Fill in the required information



[5] Click “Create”



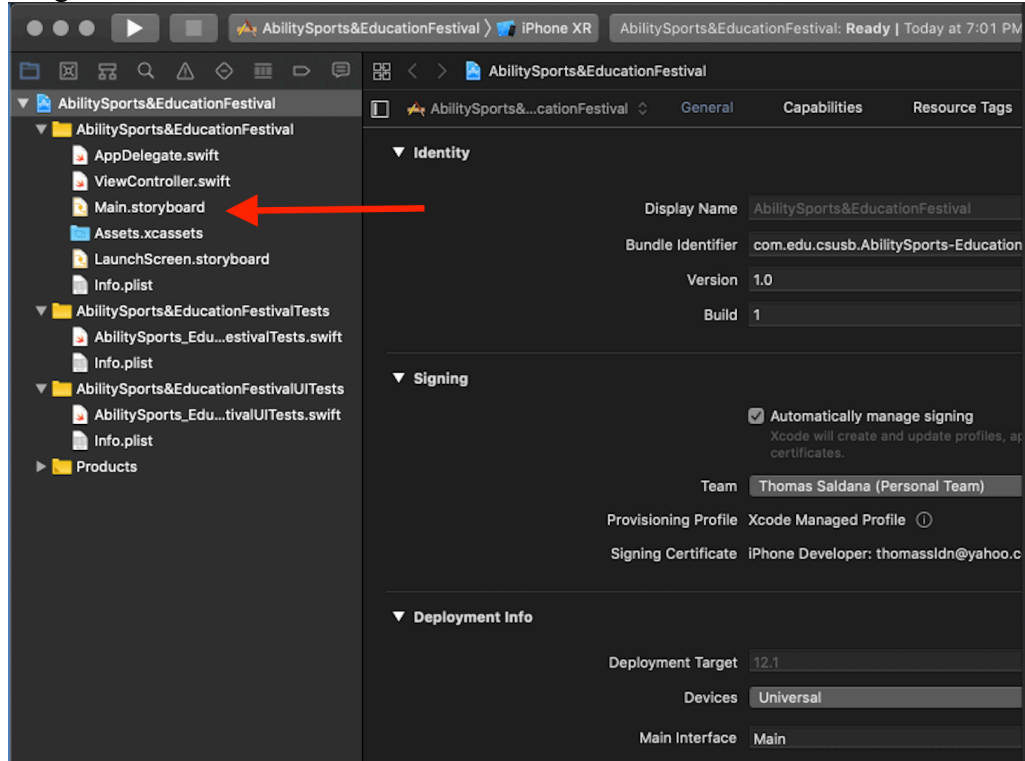
End of Section: At this point you have created the initial Xcode project.

Create Side Menu

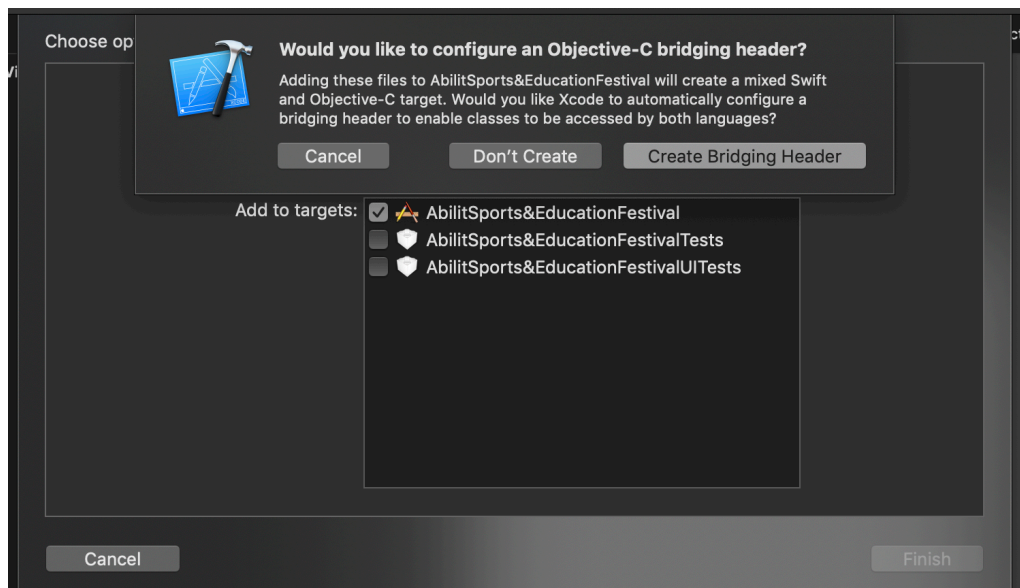
[1] Take the following two files from finder and drag them into the AbilitySports&EducationFestival folder. This will be the primary code for the Side menu. All together it is a little over 2000 lines of code.

	SWRevealViewController.h	Jun 18, 2015 at 12:20 PM
	SWRevealViewController.m	Jun 18, 2015 at 12:20 PM

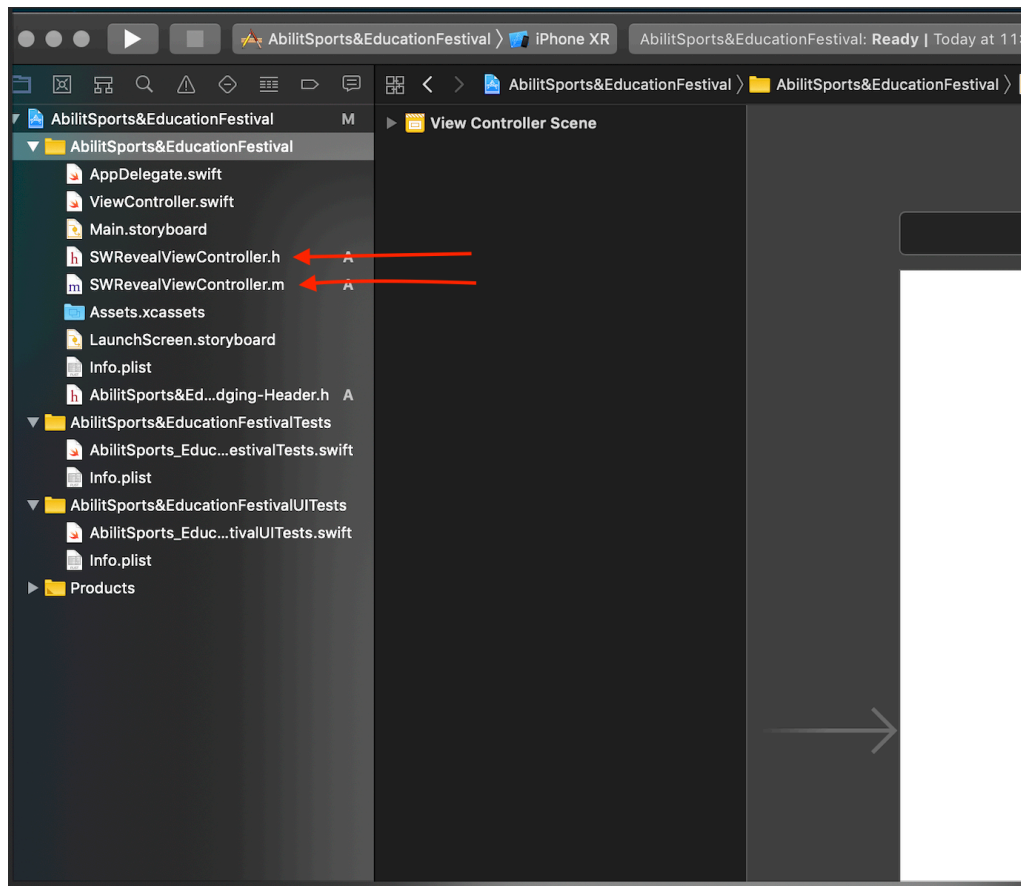
Drag the files here:



[2] Click *Finish* and then *Create Bridge Header*

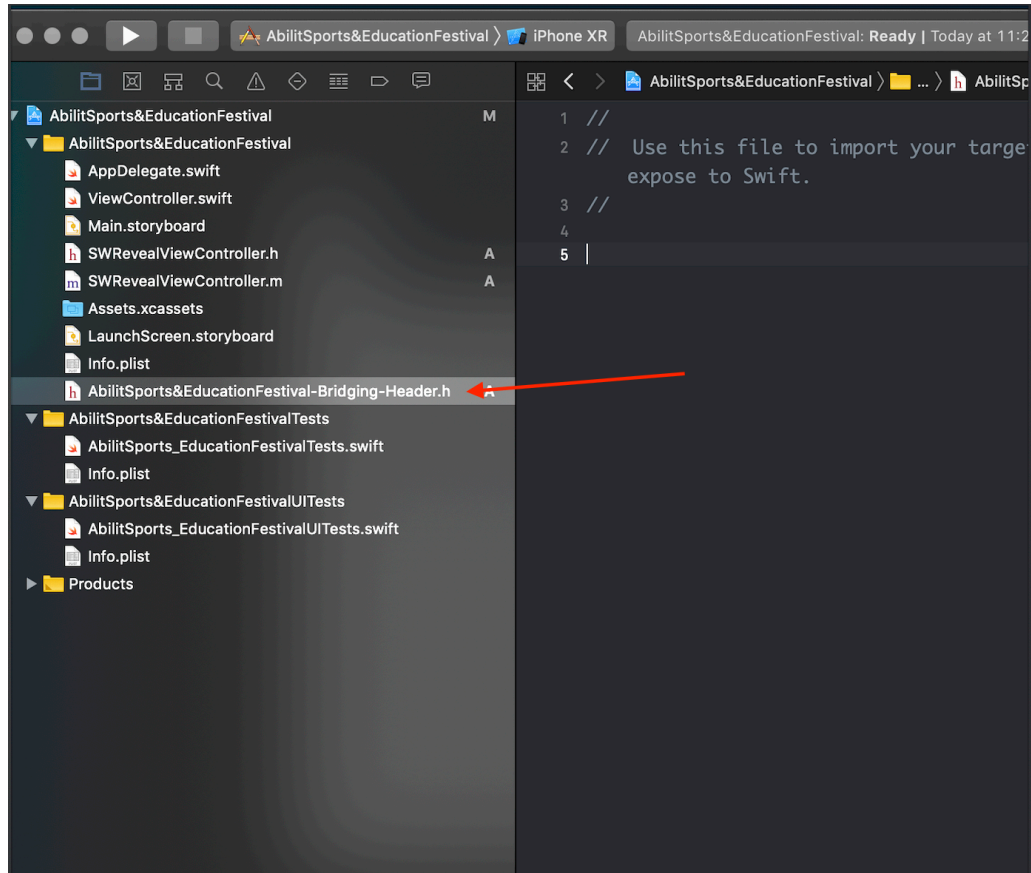


And you should now see both of your files in your project.



[3] The two files that we added to the project contain the code that will create the side menu (or “hamburger”) that is going to be used throughout the application.

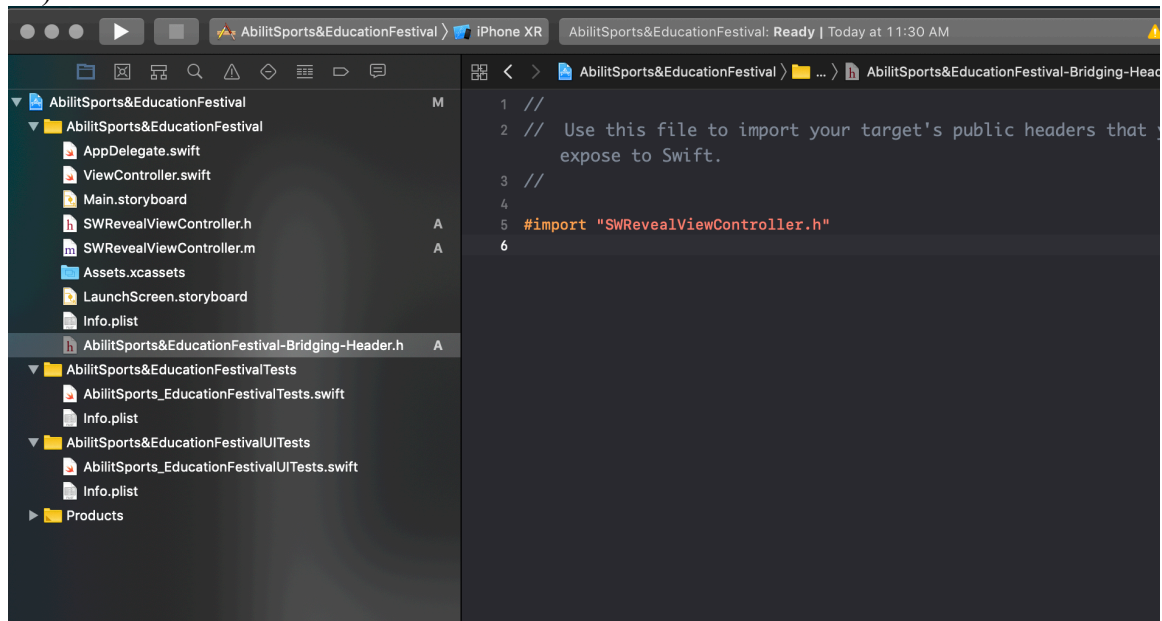
[4] Click on the *AbilitySports&EducationFestival-Bridging-Header.h* file to open it.



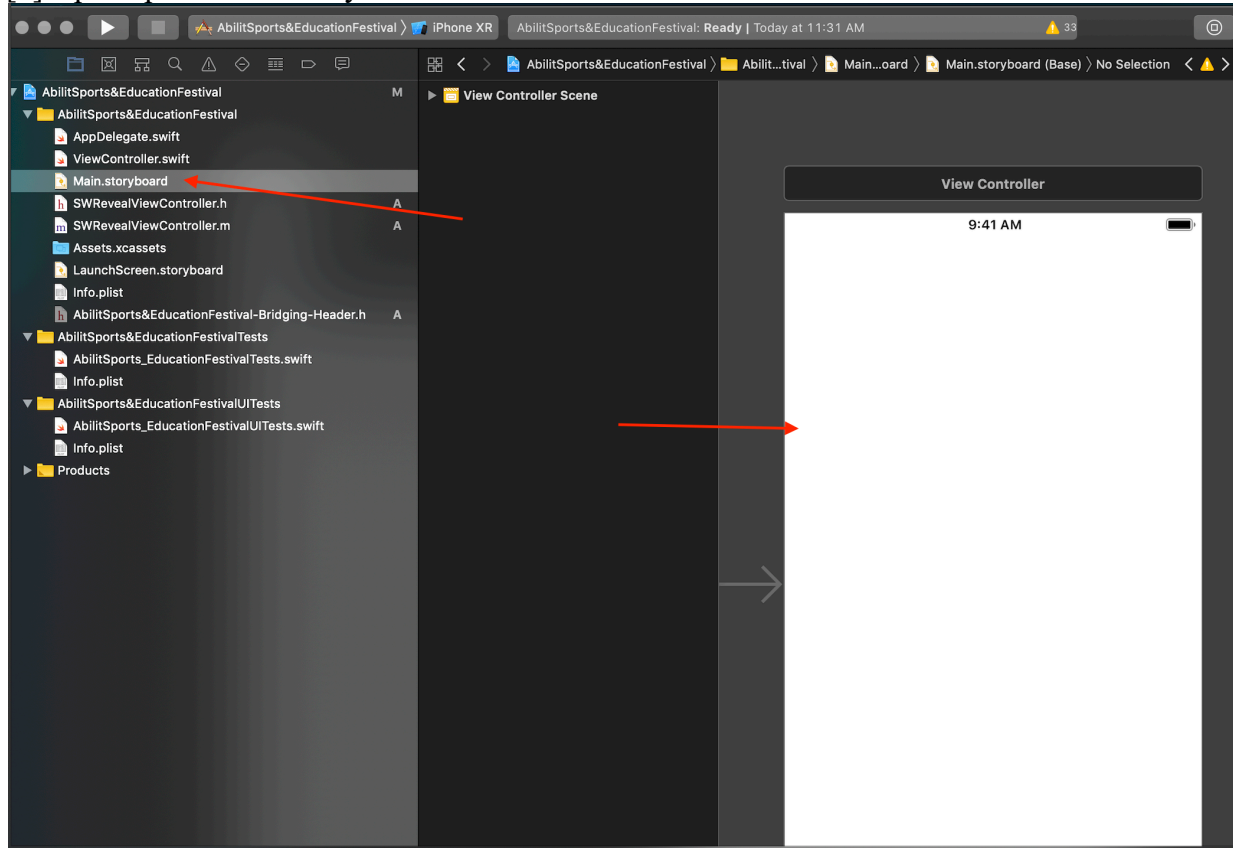
[5] Add the following line of code to the file:

```
#import "SWRevealViewController.h"
```

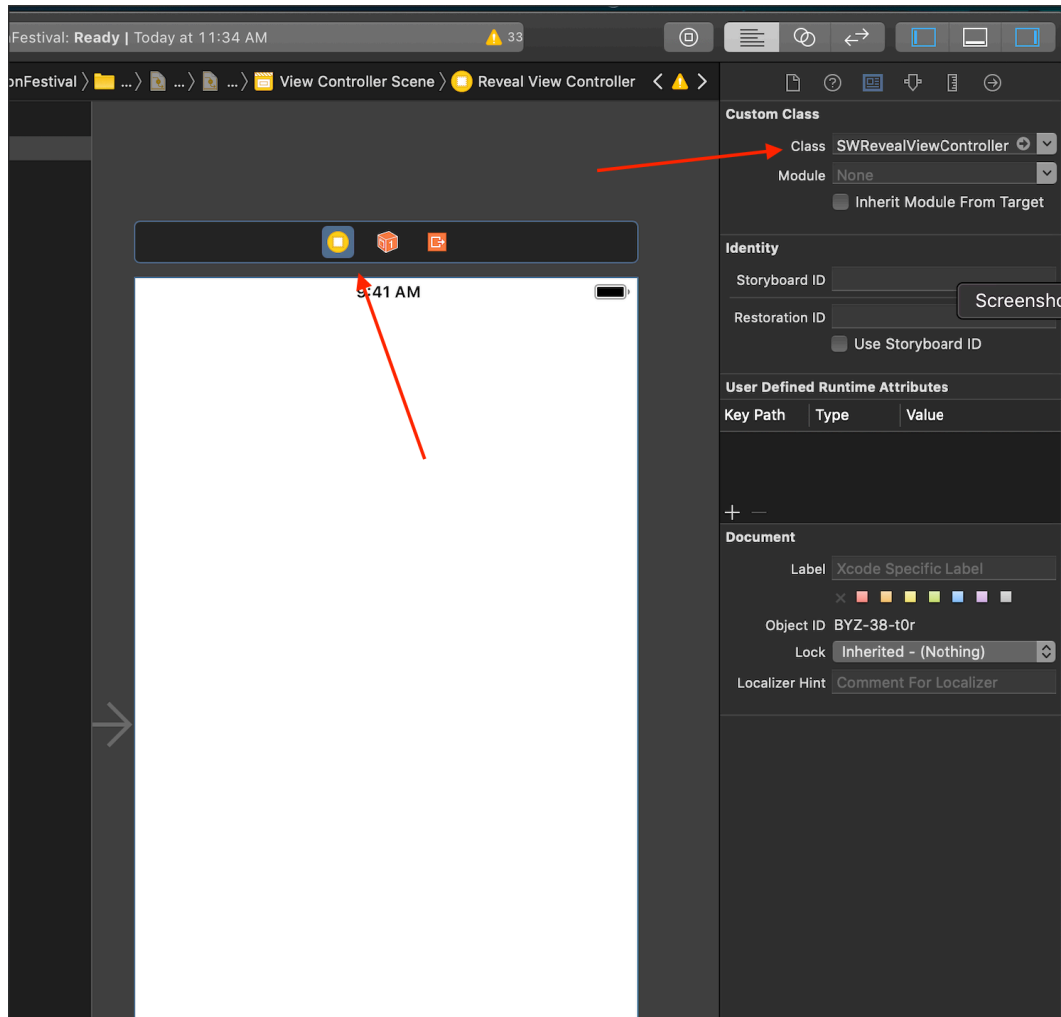
Ex)



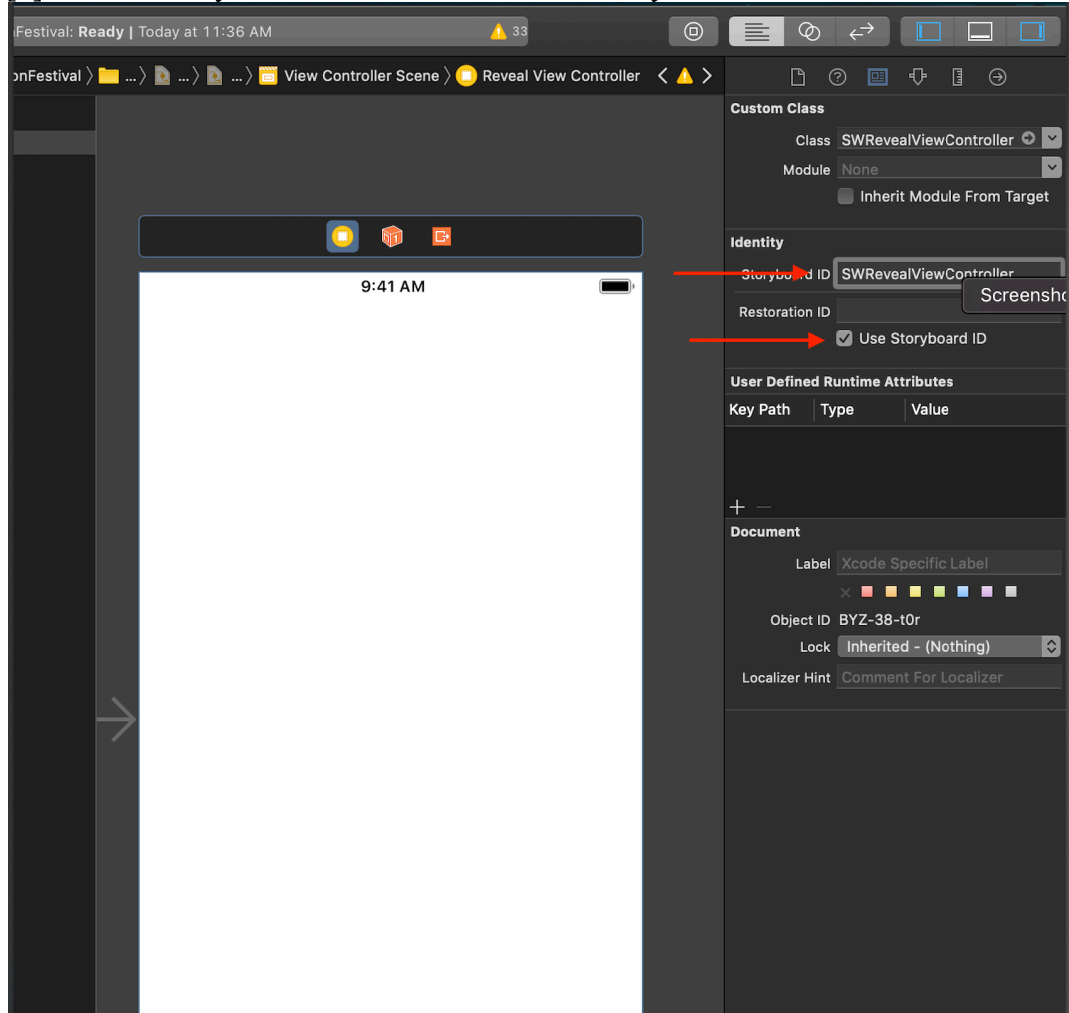
[6] Open up the Main.Storyboard



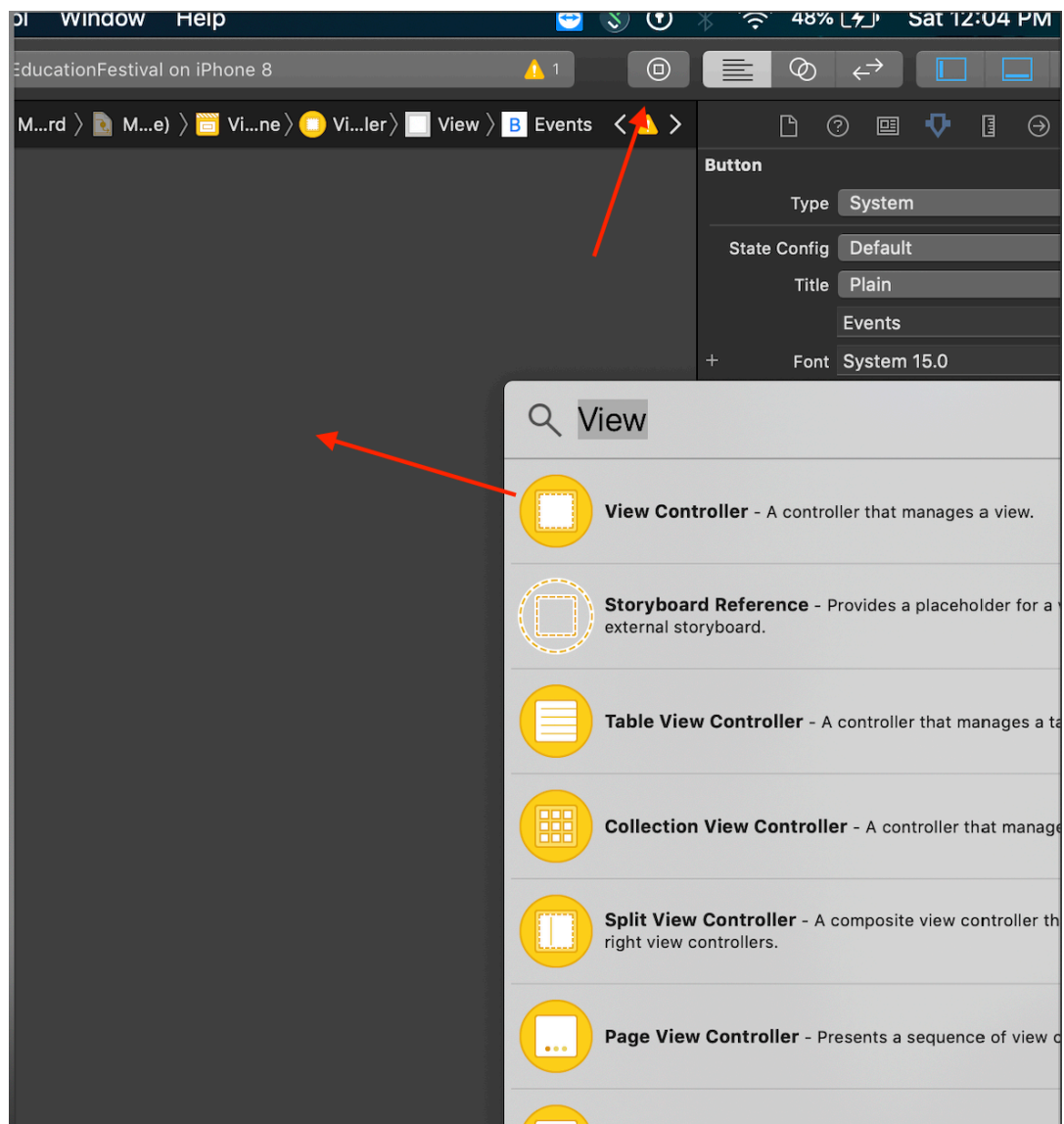
[7] Click on the View Controller and associate the View Controller with the class *SWRevealViewController.h*.



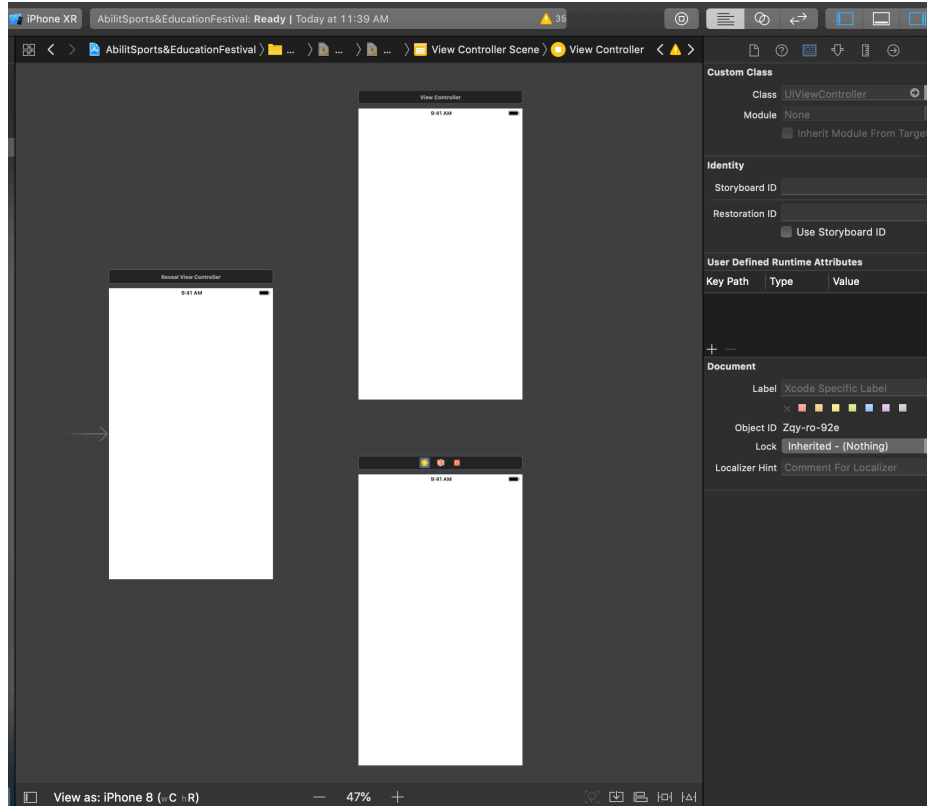
[8] Add the Storyboard ID and click on *“Use Storyboard ID”*



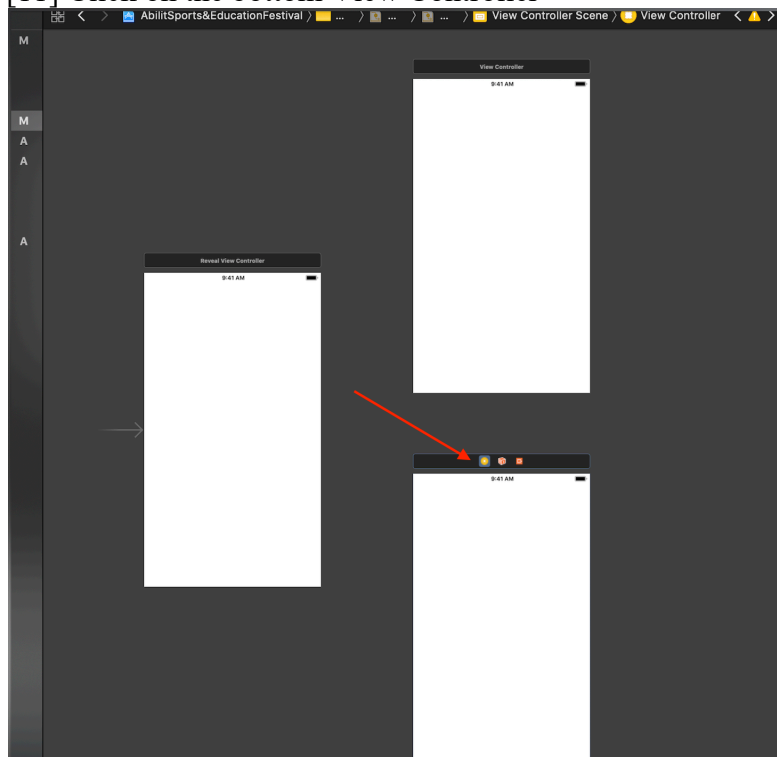
[9] Drag two new View Controllers onto the Main.storyboard.



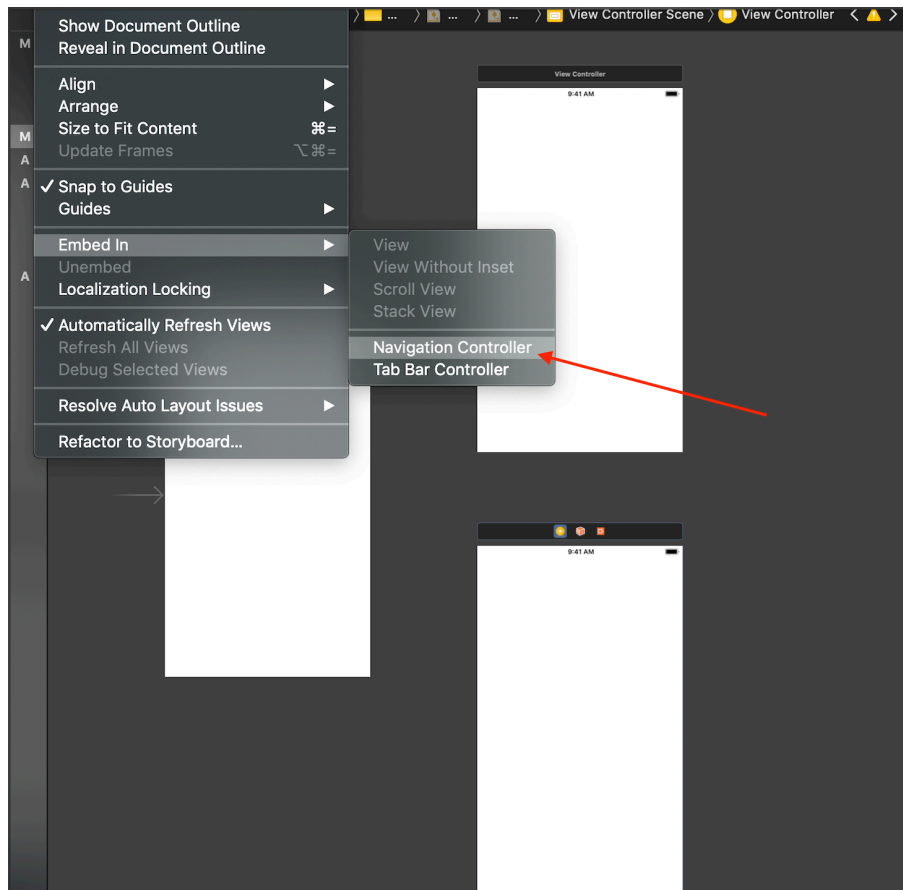
[10] After dragging the two new View Controllers onto the Main.storyboard your storyboard should look like this:



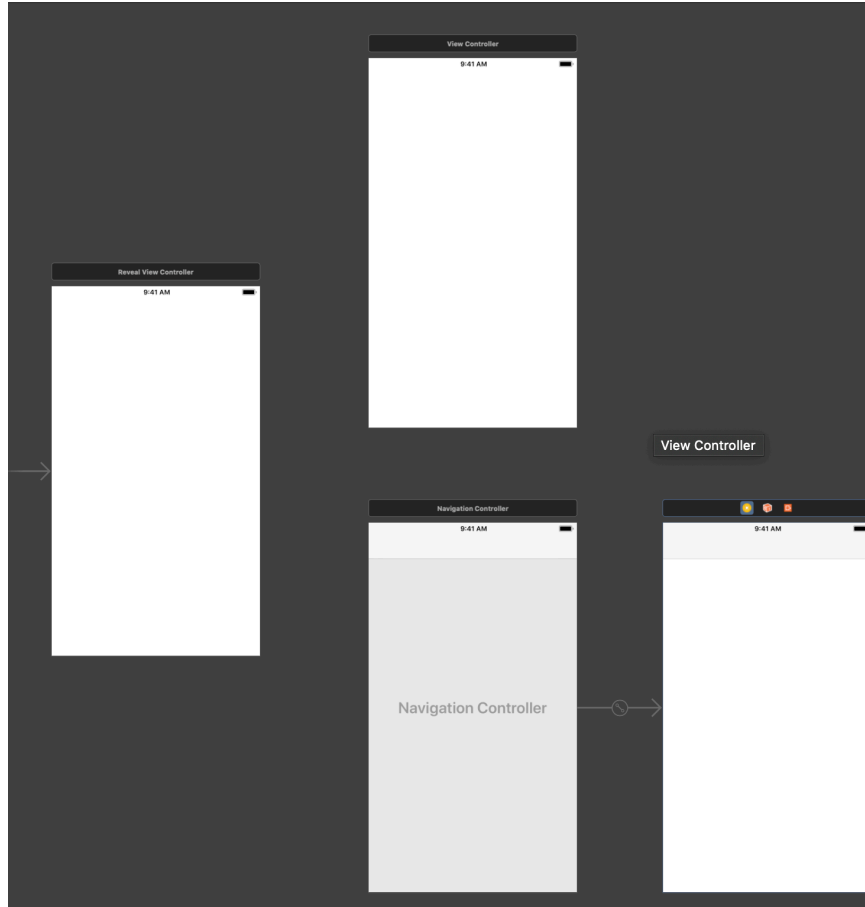
[11] Click on the bottom View Controller



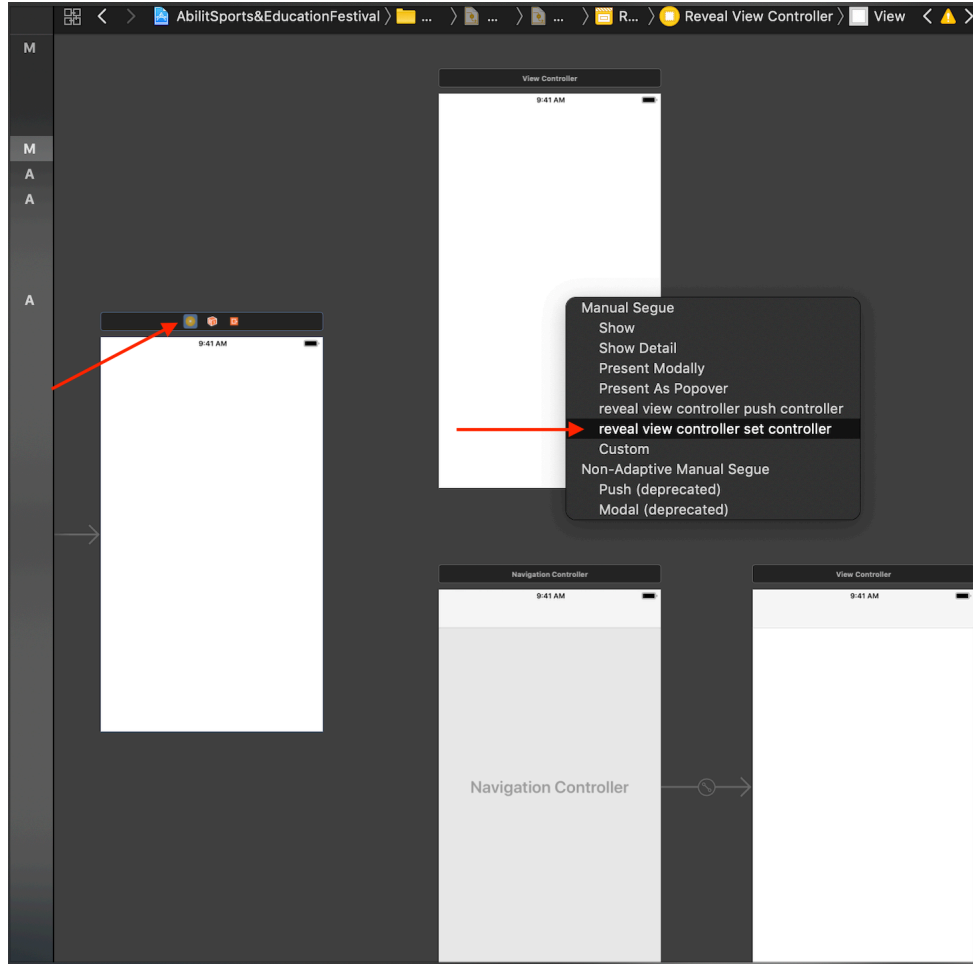
[12] Go to **Editor > Embed in > Navigation Controller** to embed the view controller into a Navigation View Controller.



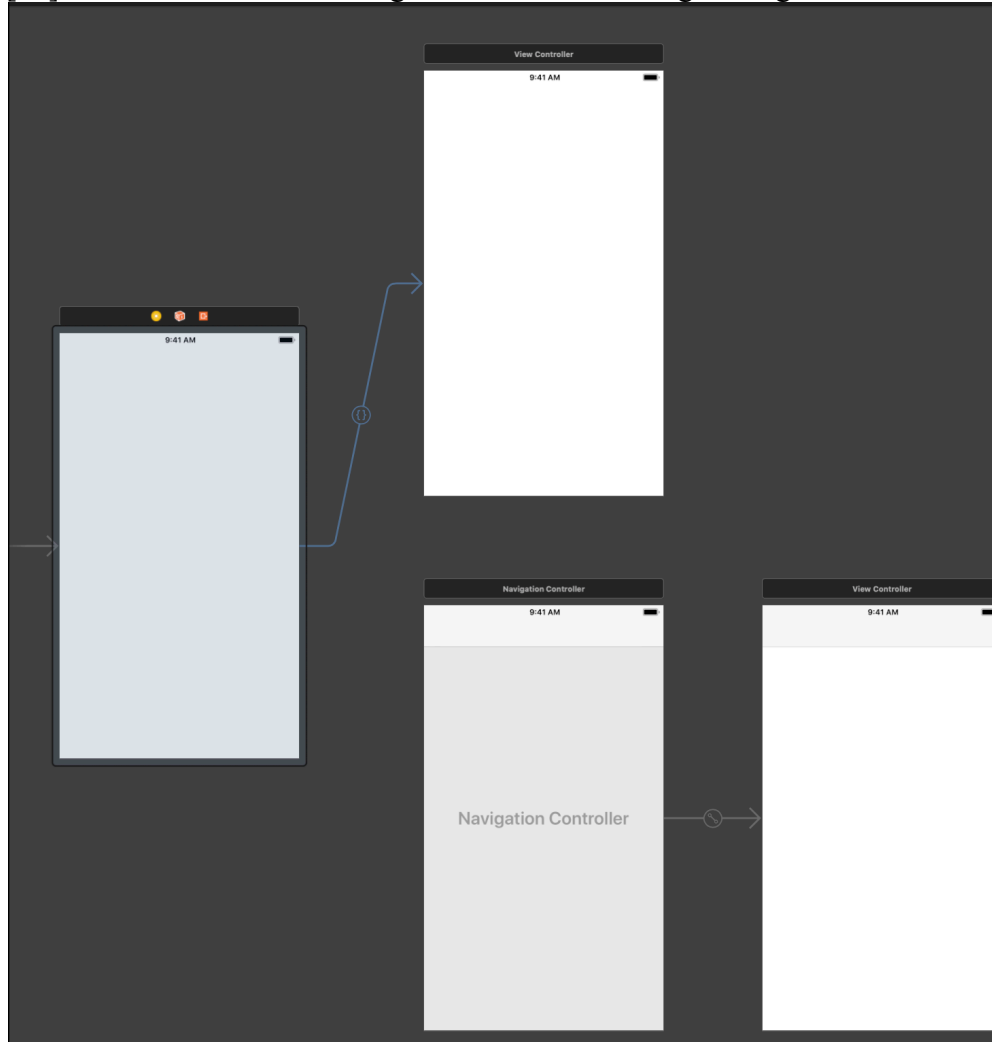
[13] At this point you should see the following in your Main.storyboard



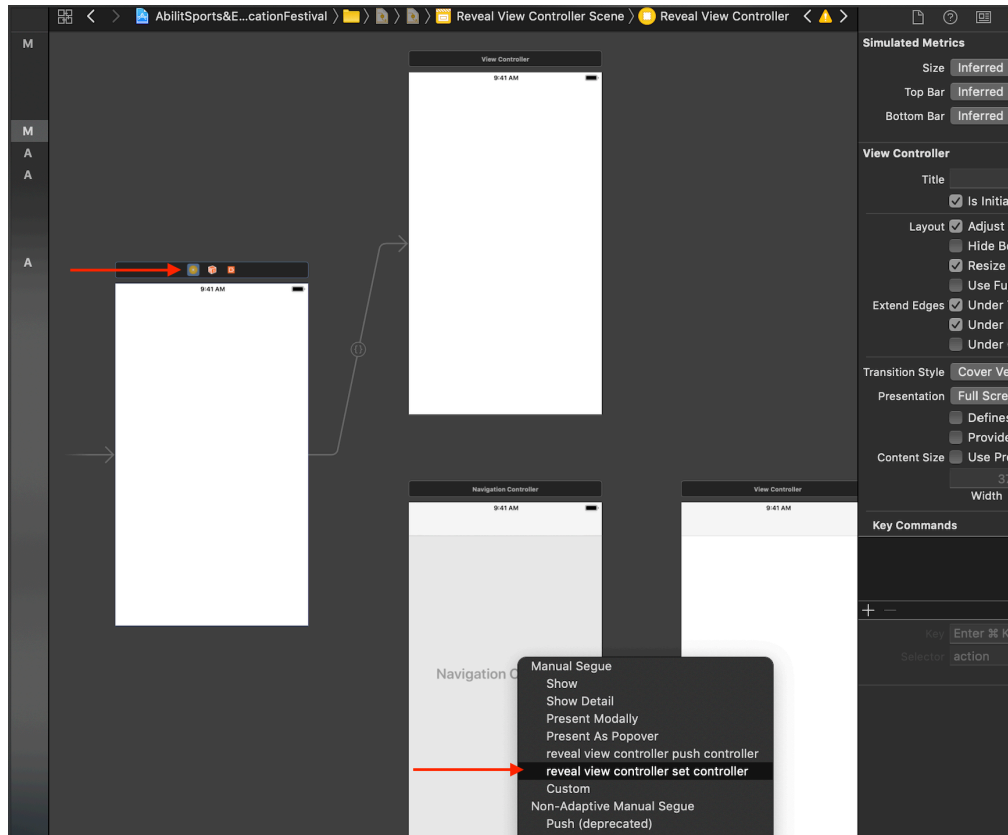
[14] Create a *segue* from the *SWRevealViewController* to one of the view controllers we just created. Choose “*reveal view controller set controller*”



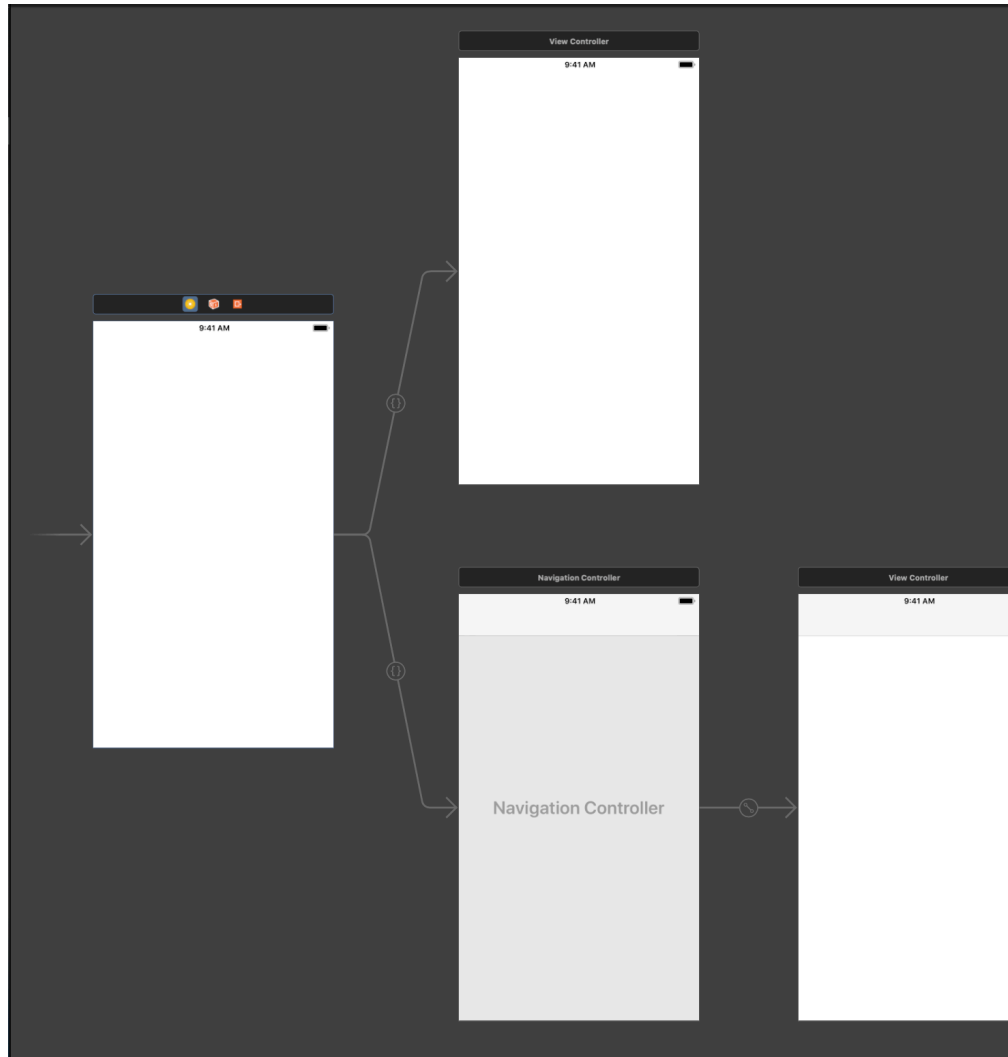
[15] You should see something like this after creating the segue:



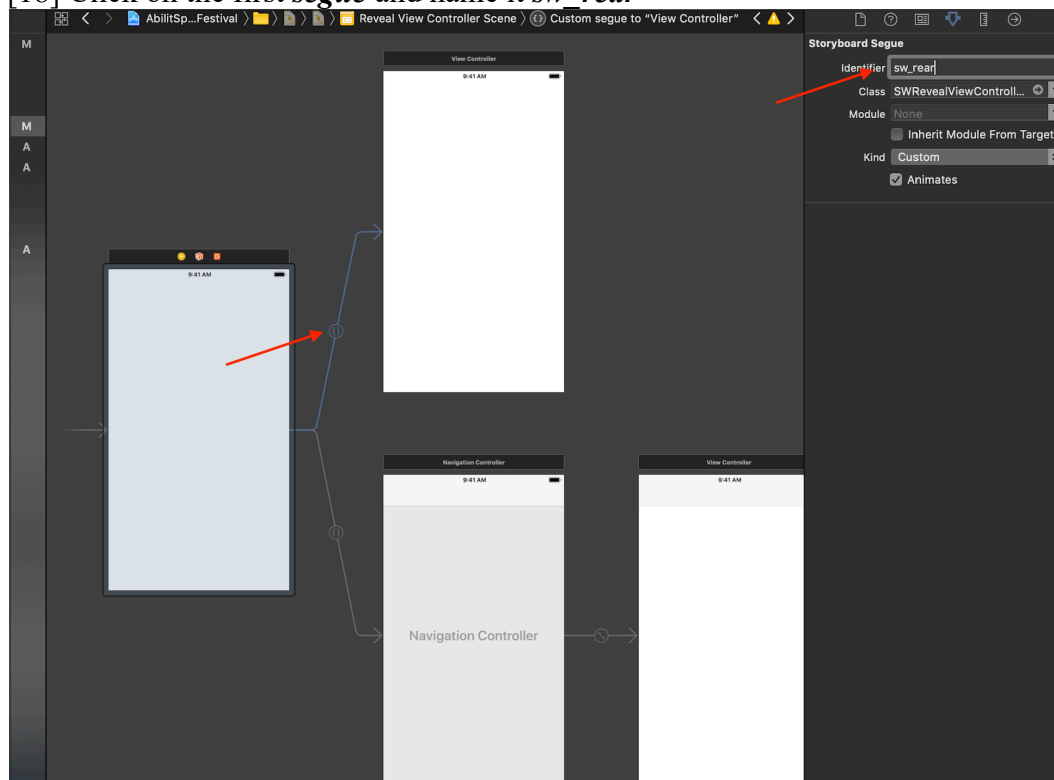
[16] Now, do the same thing, and create a *segue* from the *SWRevealViewController* to the other view controller we just created that has the Navigation view controller. Choose “reveal view controller set controller”



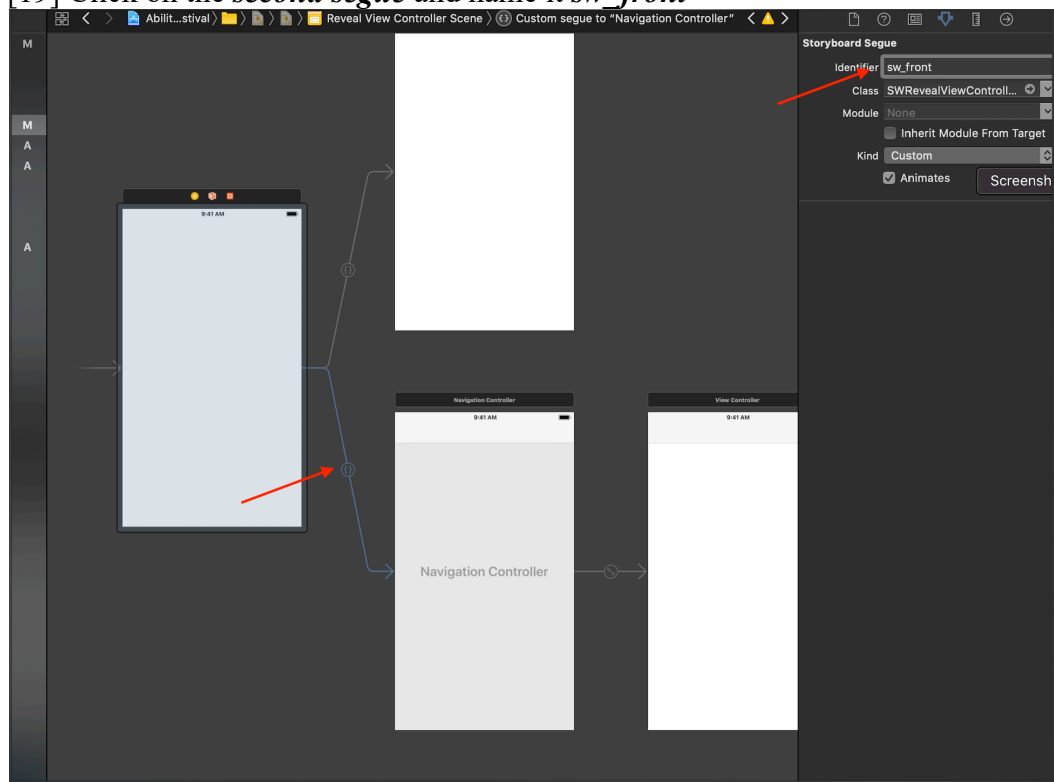
[17] Now, you should see something like this:



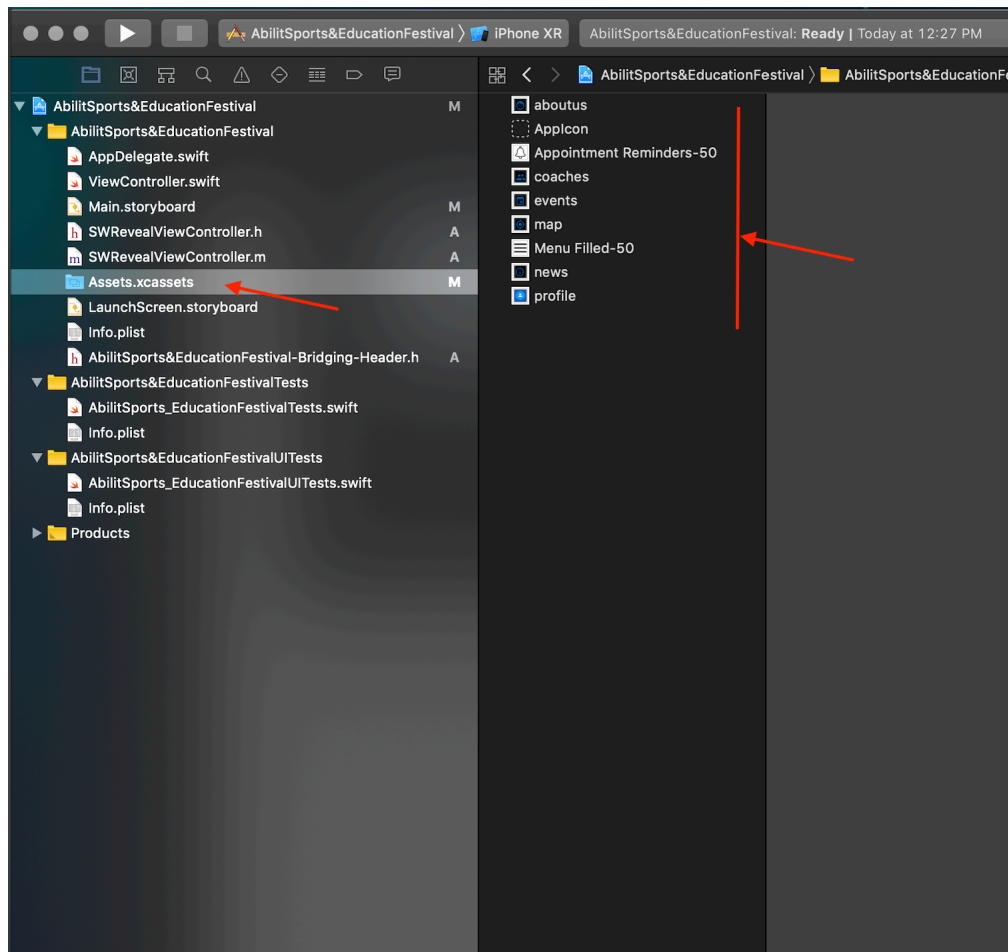
[18] Click on the first *segue* and name it *sw_rear*



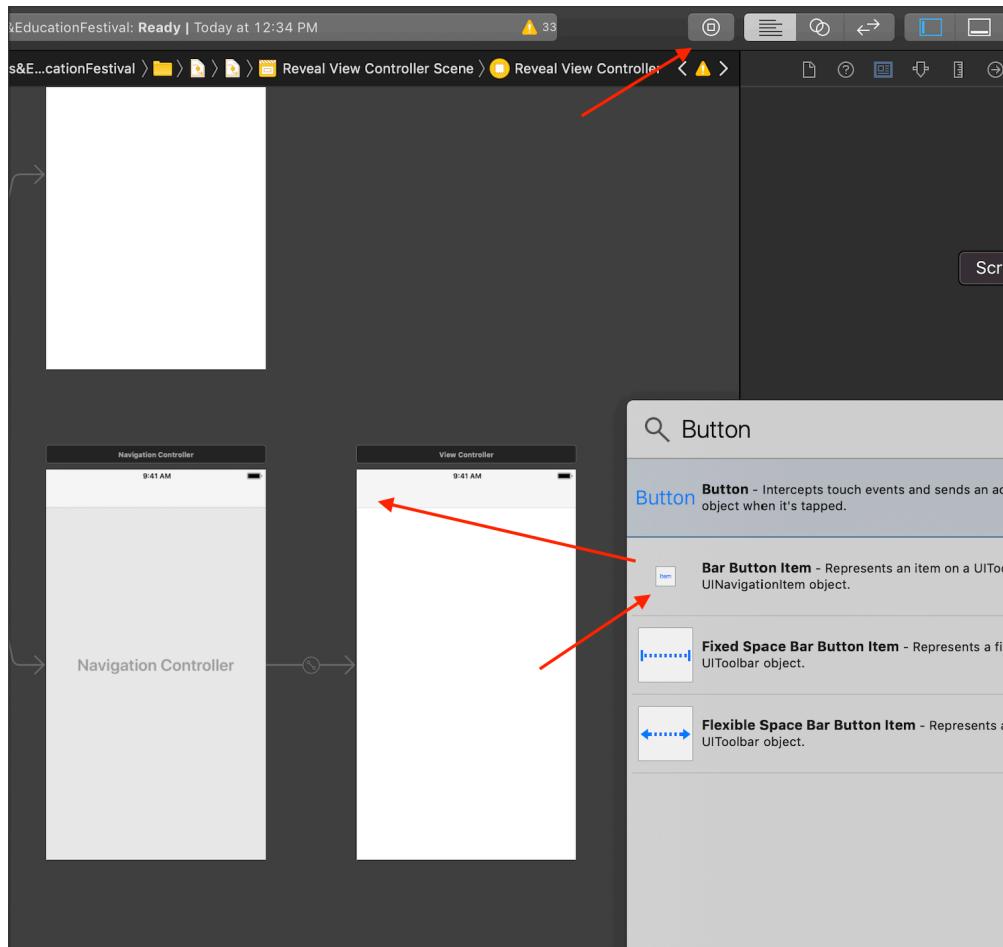
[19] Click on the *second segue* and name it *sw_front*



[20] Now that that is complete, we will create a navigation button on the tab bar so that we can open up the side menu. But first, we have to import some images into the project. Import the following images into the project. We will use the “*Menu Filled – 50*” image next.

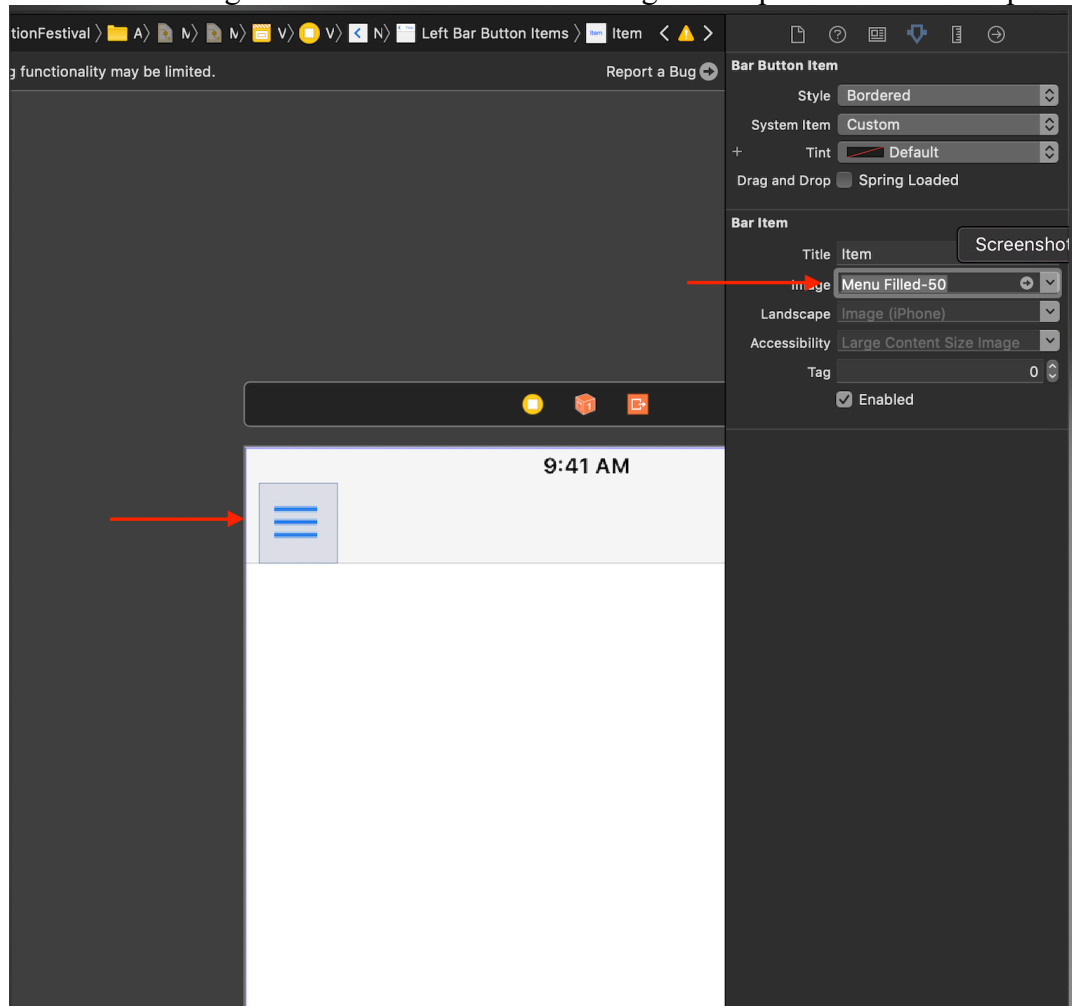


[21] Drag a Bar Button item onto the Toolbar

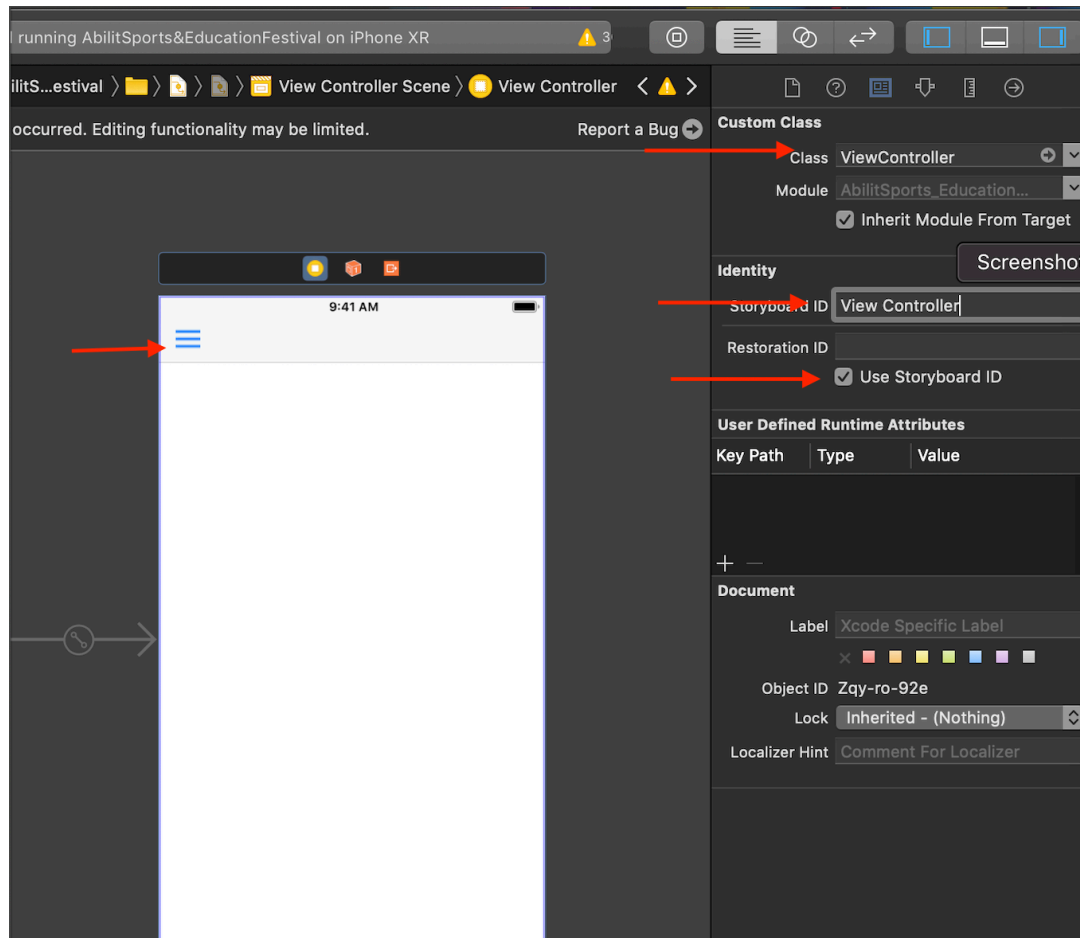


[22] Make sure to add constraints. You should see something like this:

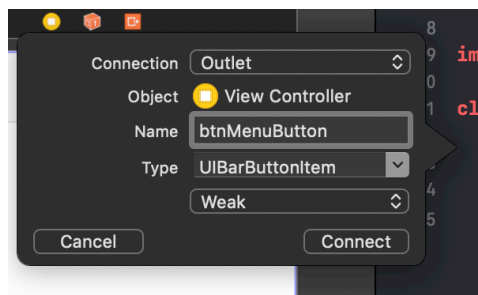
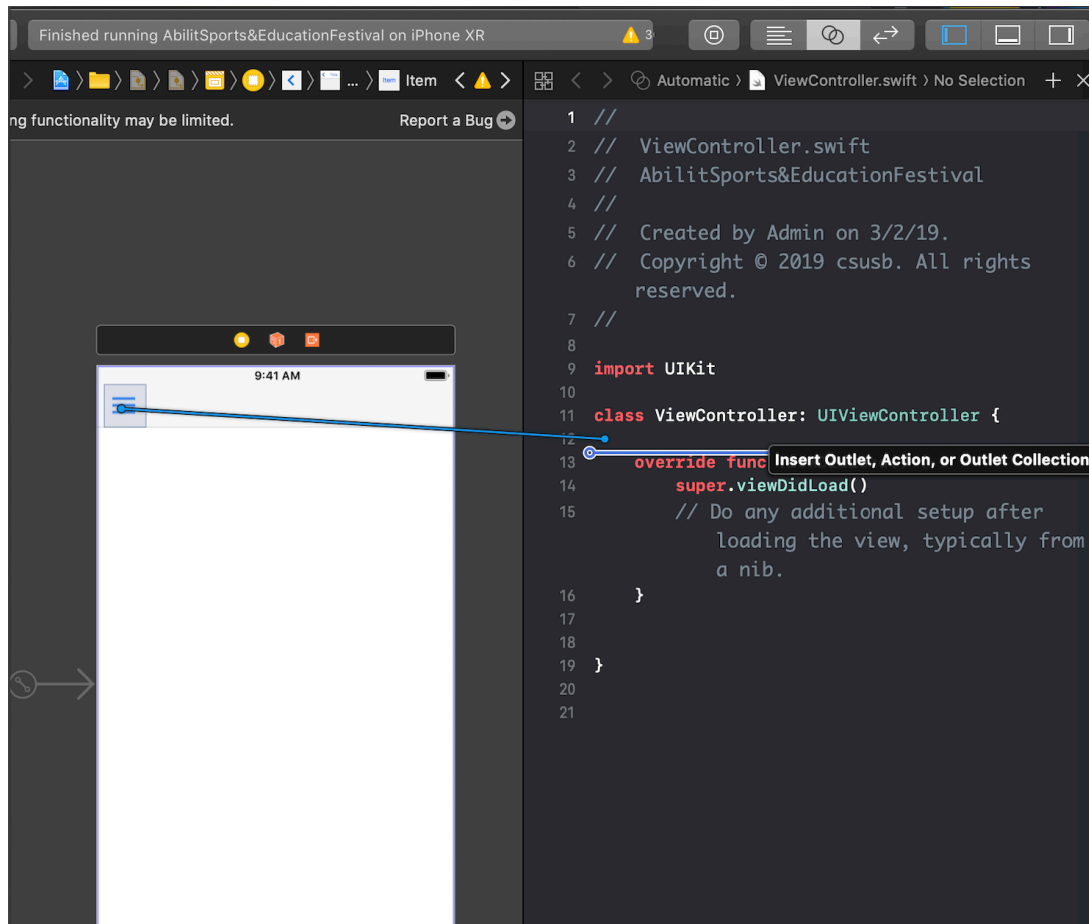
Note: set the image of the button to the menu image we imported in earlier steps



[23] Associate the View controller with the View Controller class



[24] Add a *variable* for the menu button in the *ViewController* class



[25] After you associate the View controller with the *ViewController.swift* class. Add the following lines to the *ViewController.swift* class inside the *viewDidLoad()* method.

```
btnMenuButton.target = revealViewController()
btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))
```

[26] Your ViewController.swift class should look something like this:

```
//
import UIKit

class ViewController: UIViewController {

    //Right Menu Button on home screen
    @IBOutlet weak var btnMenuButton: UIBarButtonItem!

    override func viewDidLoad() {
        super.viewDidLoad()

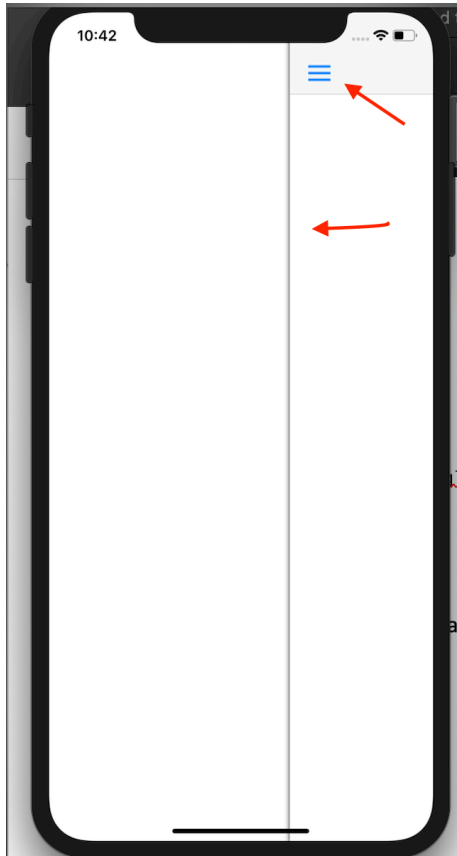
        btnMenuButton.target = revealViewController()
        btnMenuButton.action = #selector(SWRevealViewController.revealToggle(_:))

    }

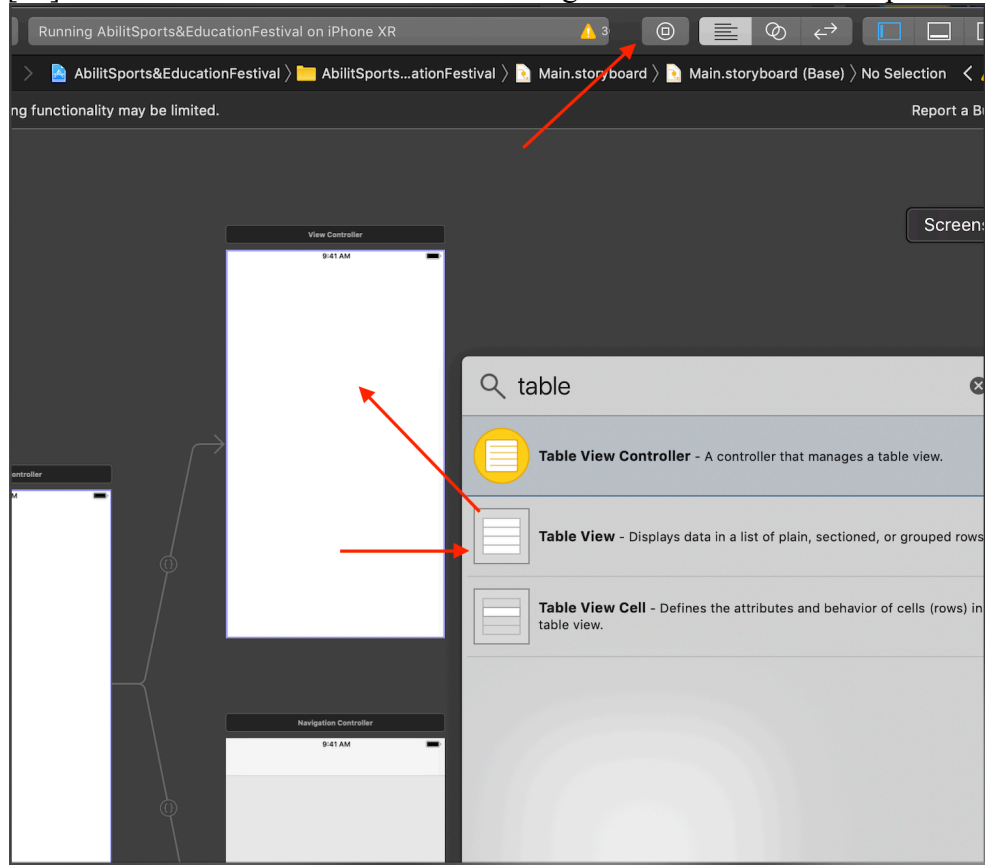
}
```


[27] After adding those lines of code, you should now be able to run the application and view the side menu when you press the side menu button.

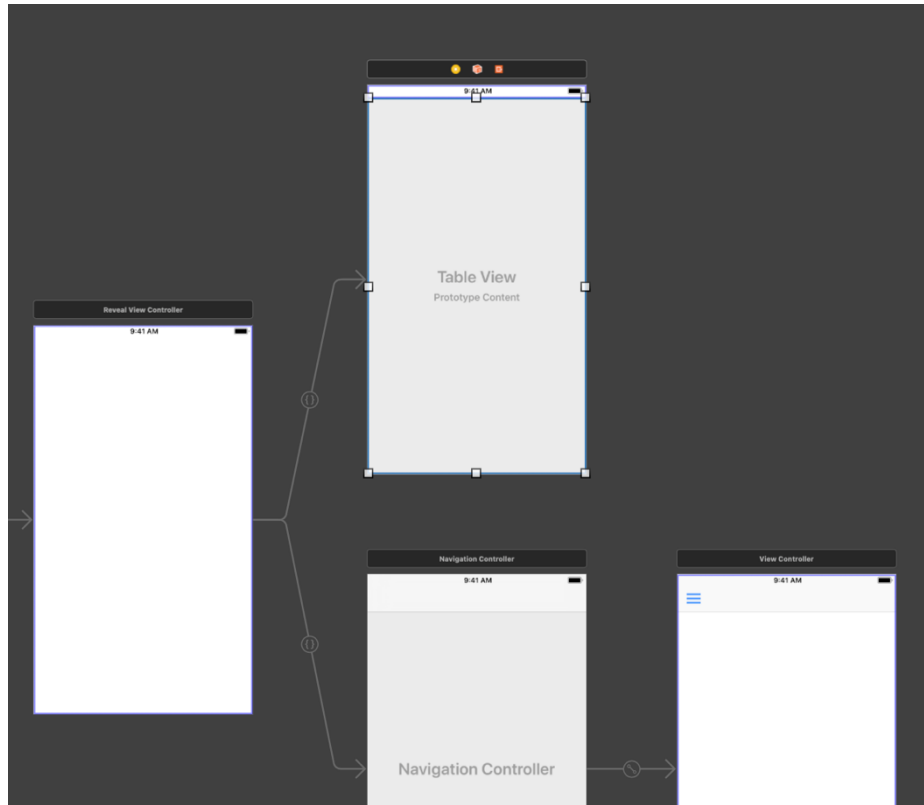
Ex)



[28] Create a table view for side menu. Drag a table view onto the top View controller

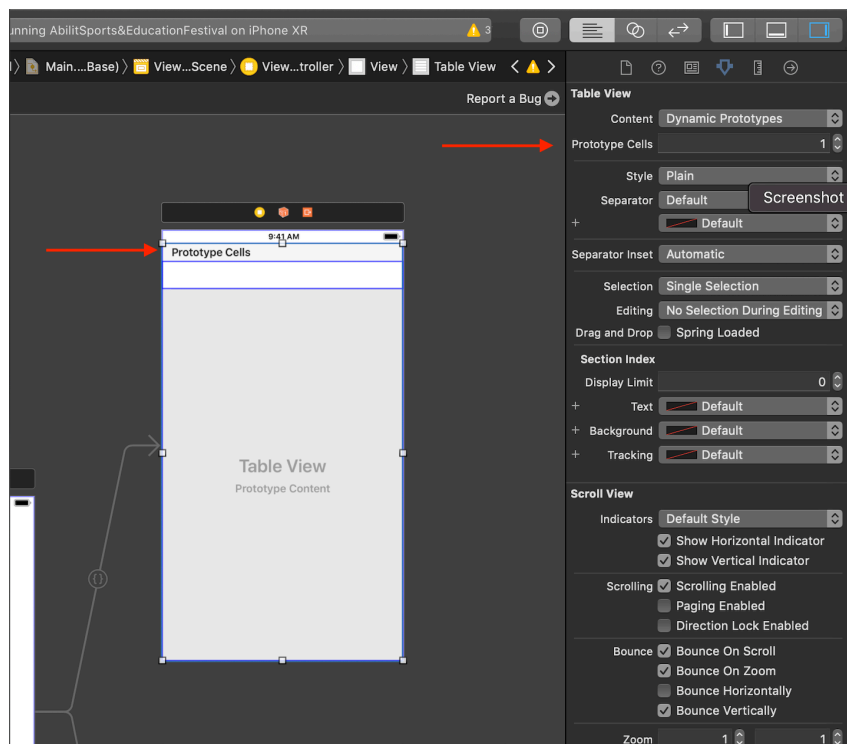


[29] Make sure the **Table View** takes up the whole screen. Add constraints.

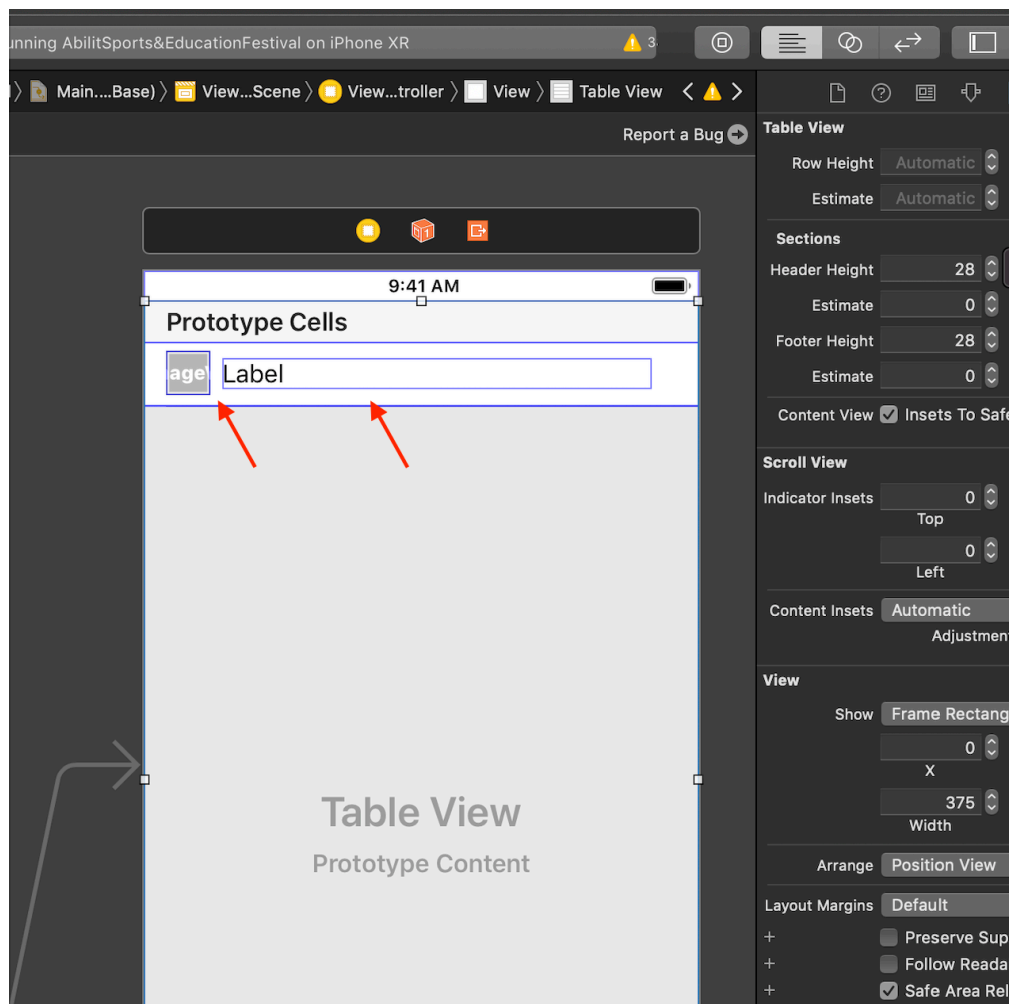


[30] Add one

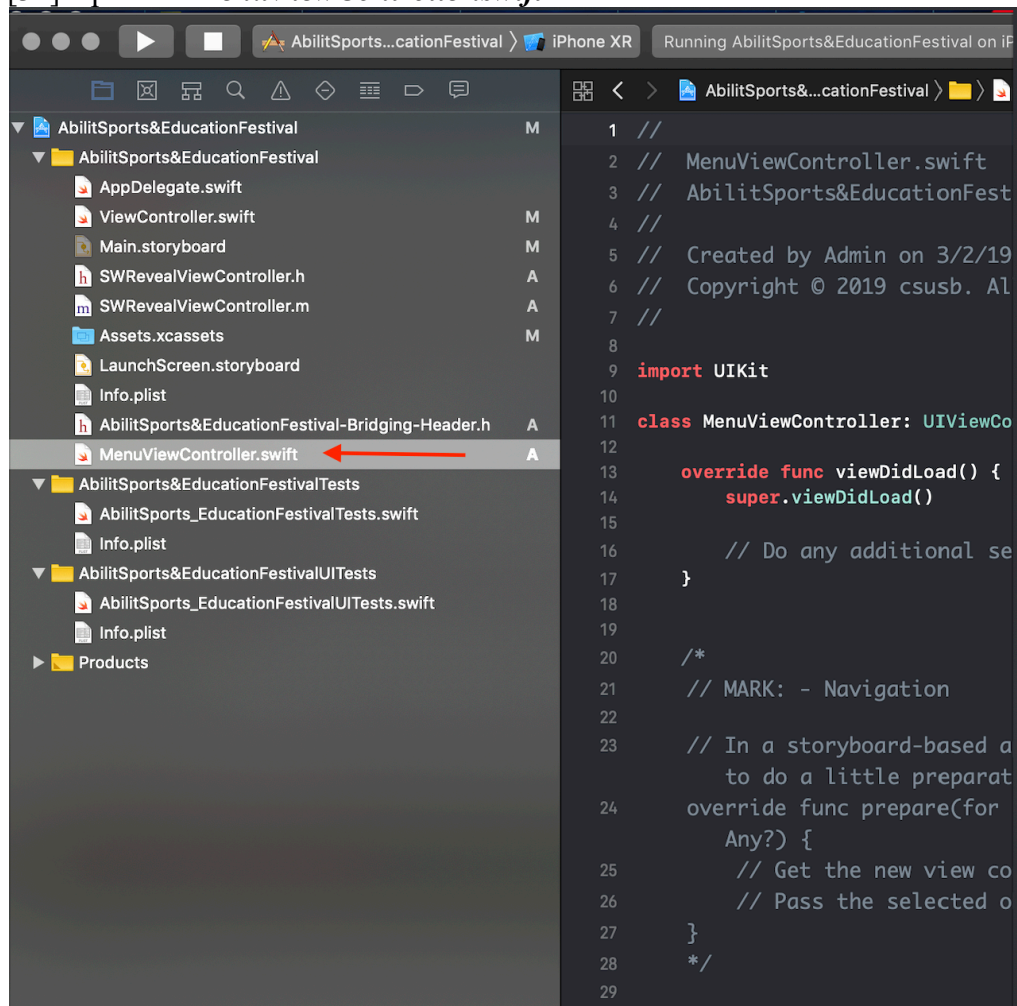
Prototype cell.



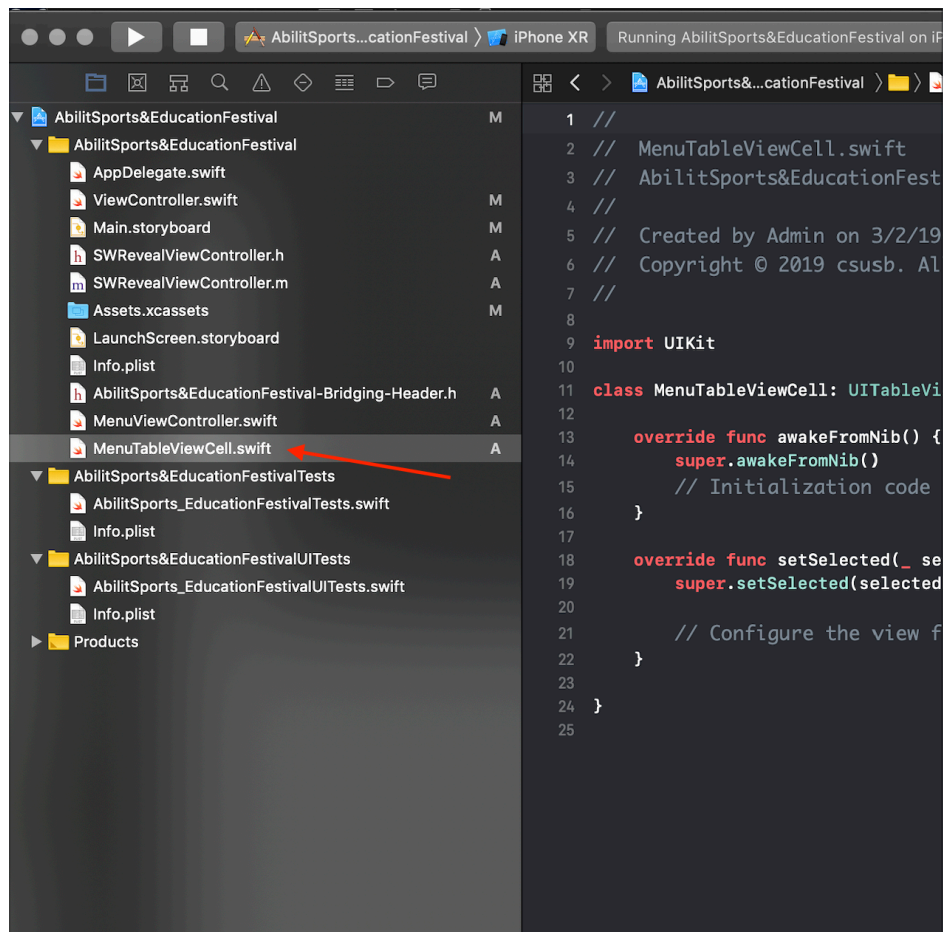
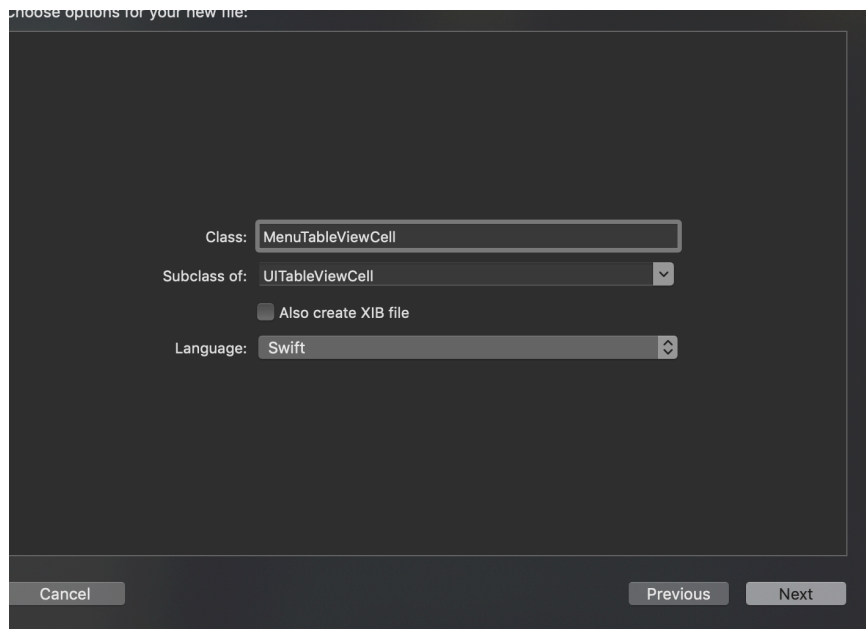
[31] In the prototype cell, drag an image and a label. This will be the image and the label for the side menu items.



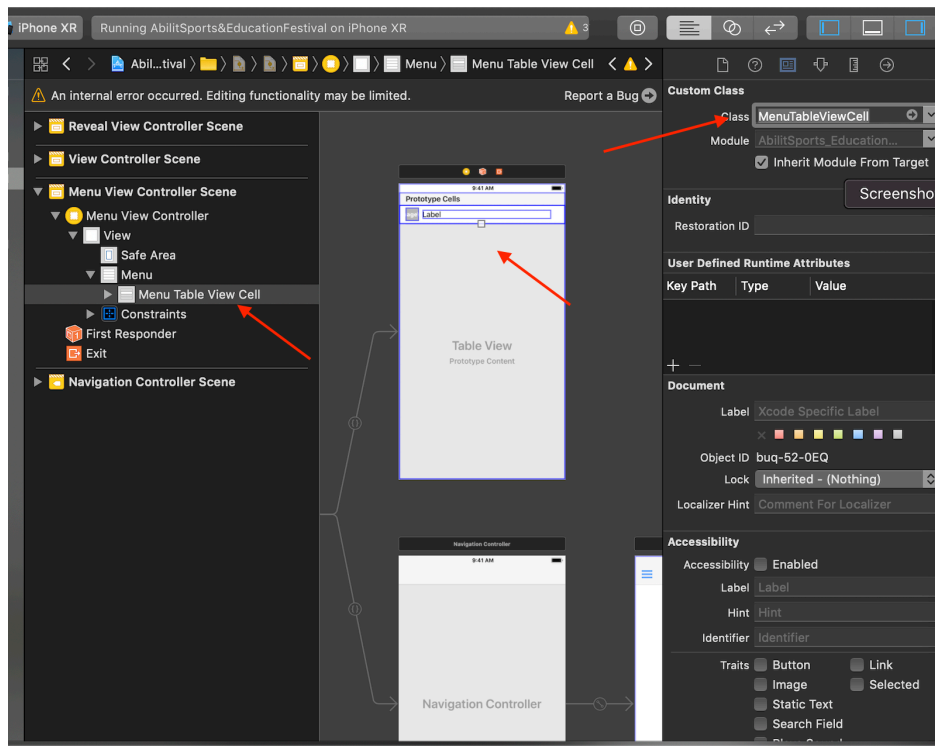
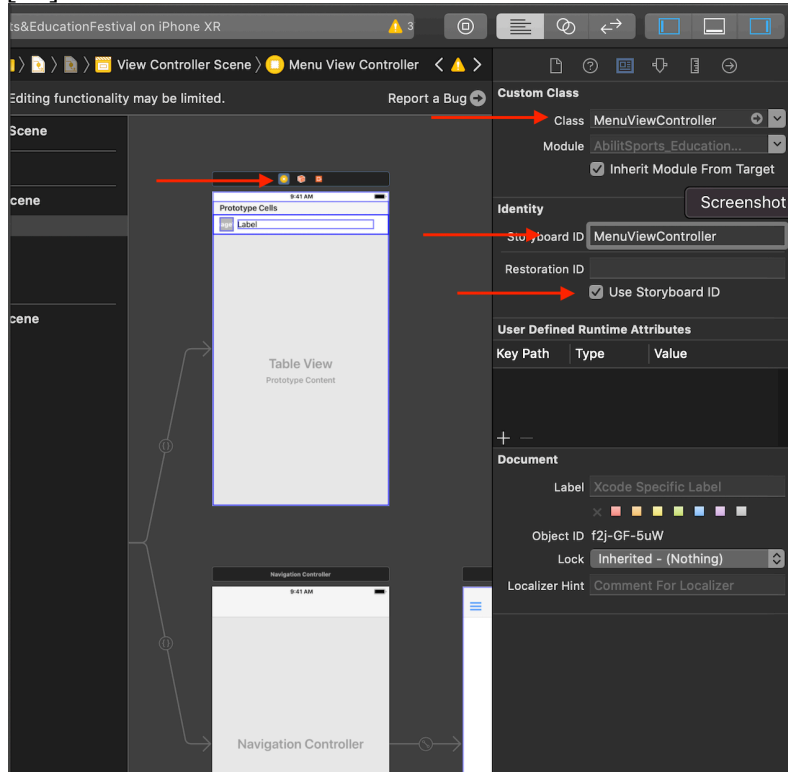
[32] Open the *MenuViewController.swift* class



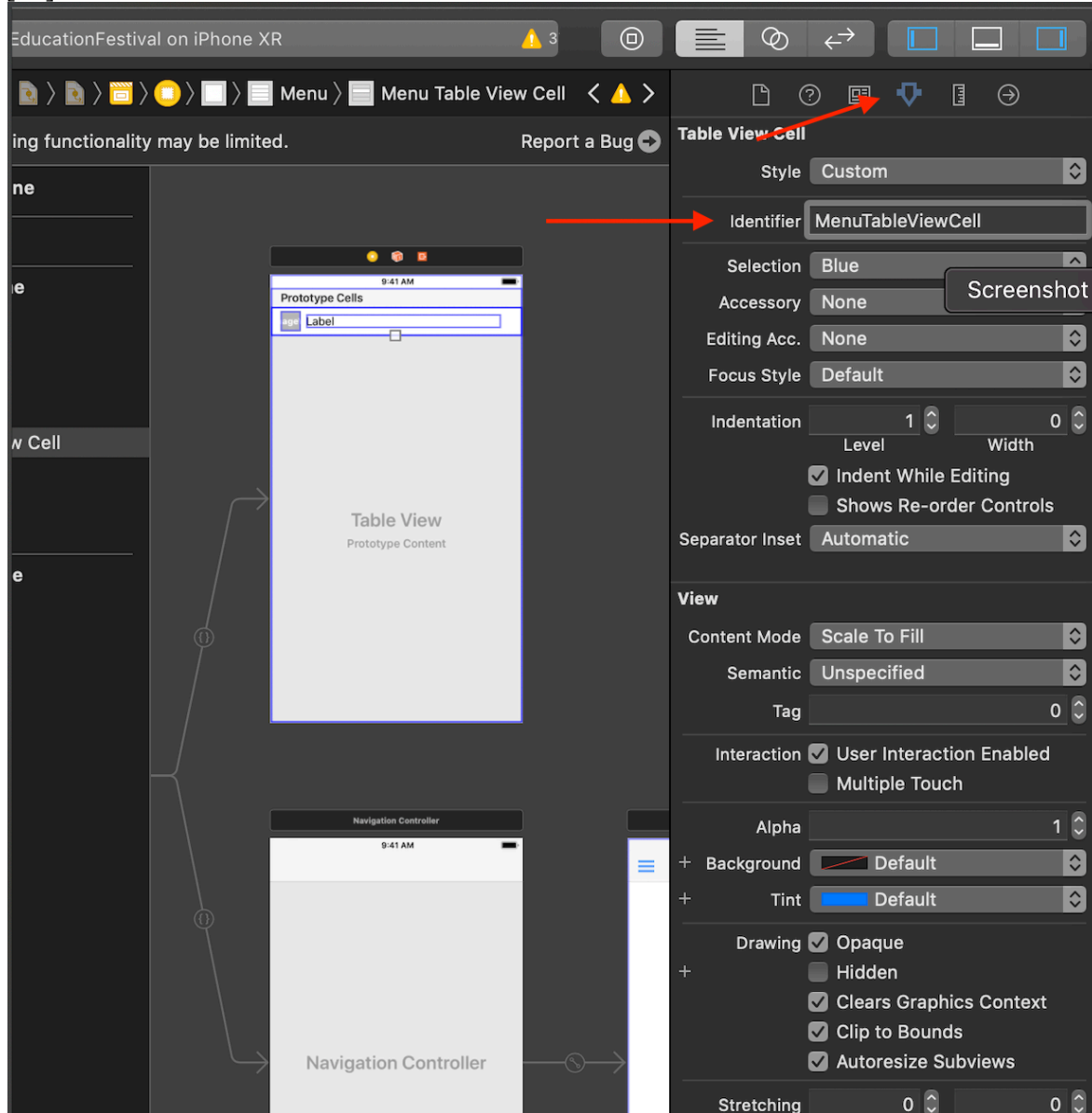
[33] Create a new class called *MenuTableViewCell* of type *UITableViewCell*



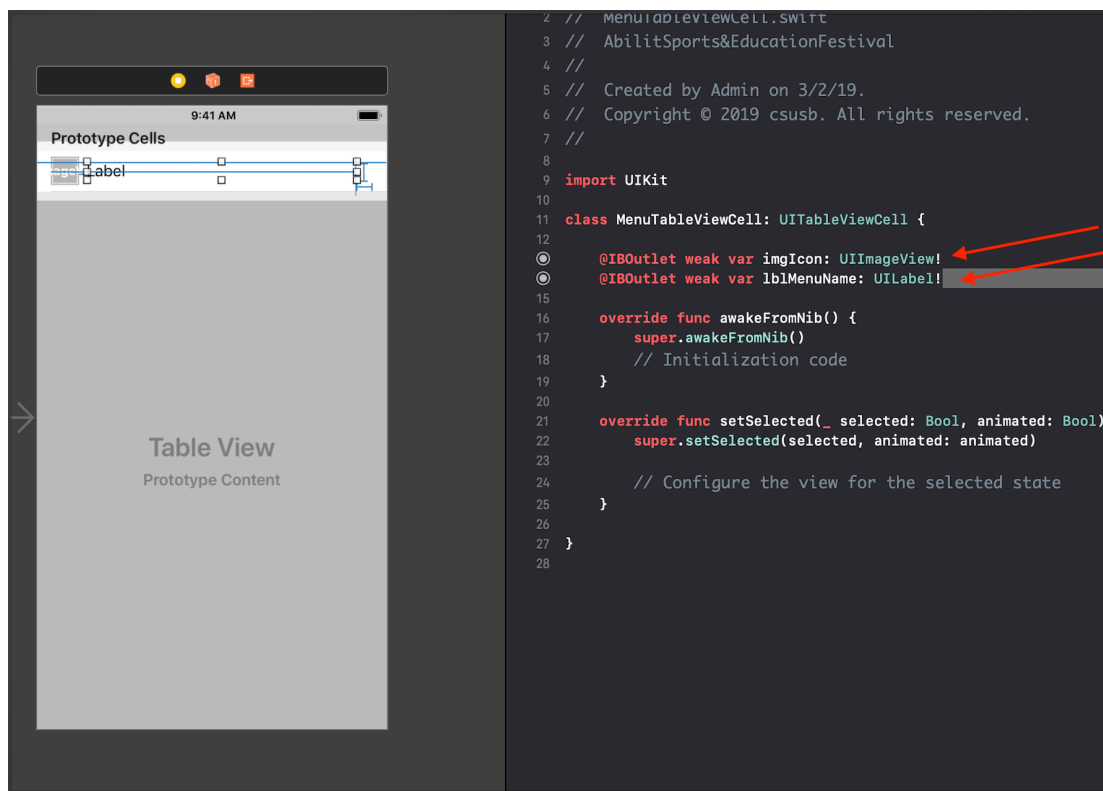
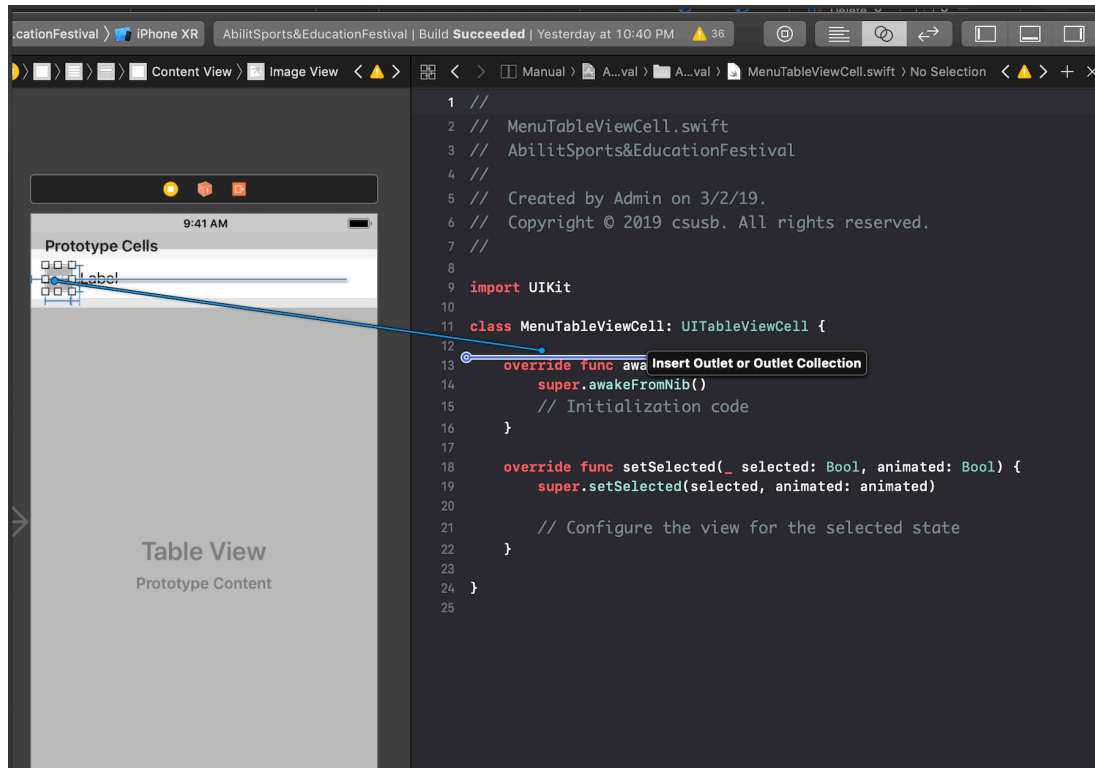
[34] Associate the *Table View* View with the *MenuViewController*



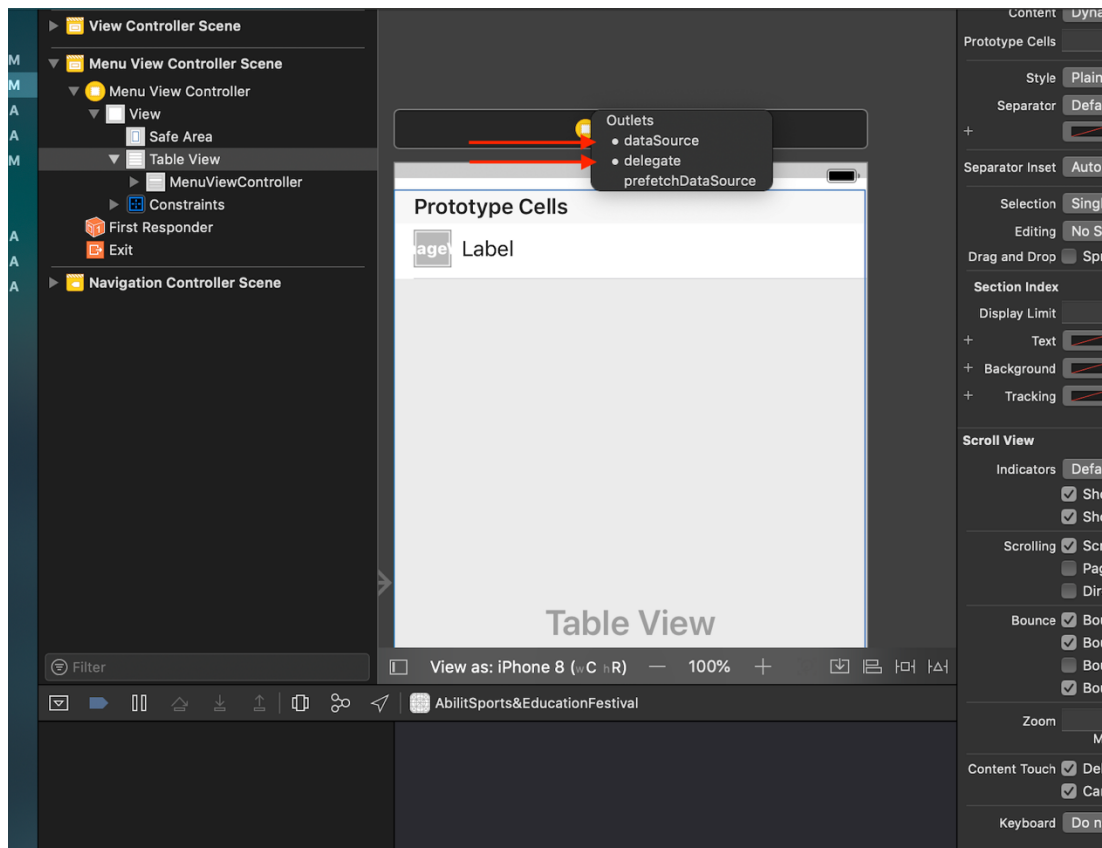
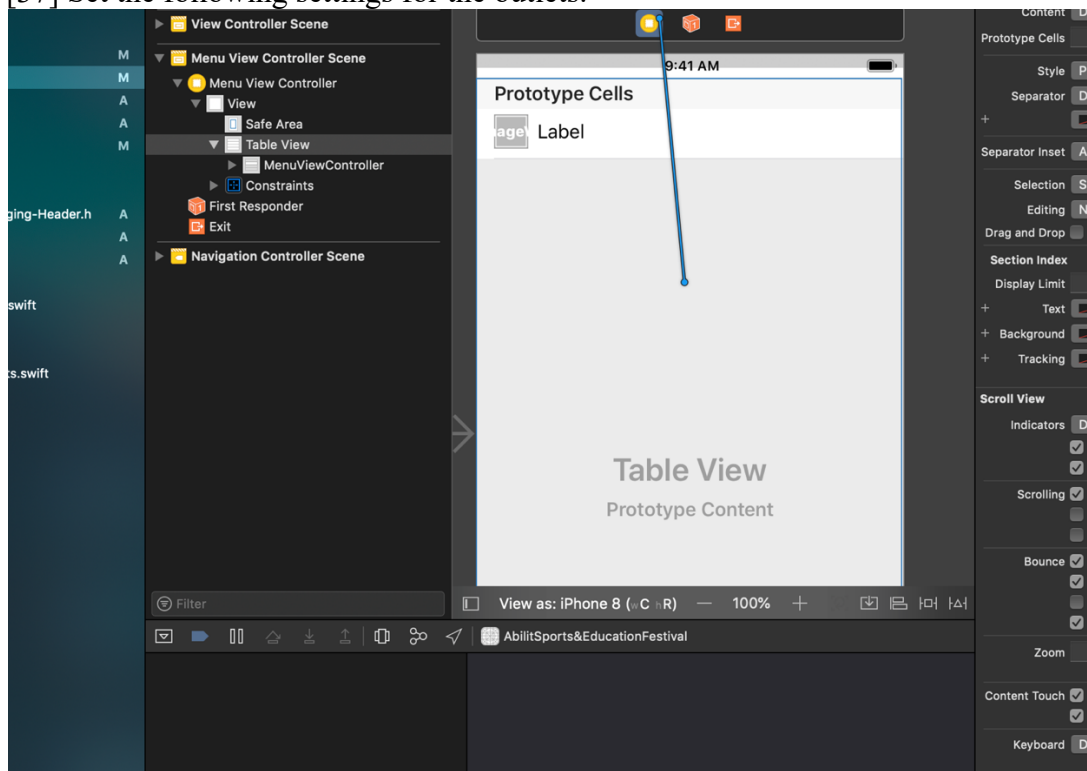
[35] Create an identifier for the *Table View*



[36] Create two variables in the MenuTableViewCell class, one for the Image and one for the label.

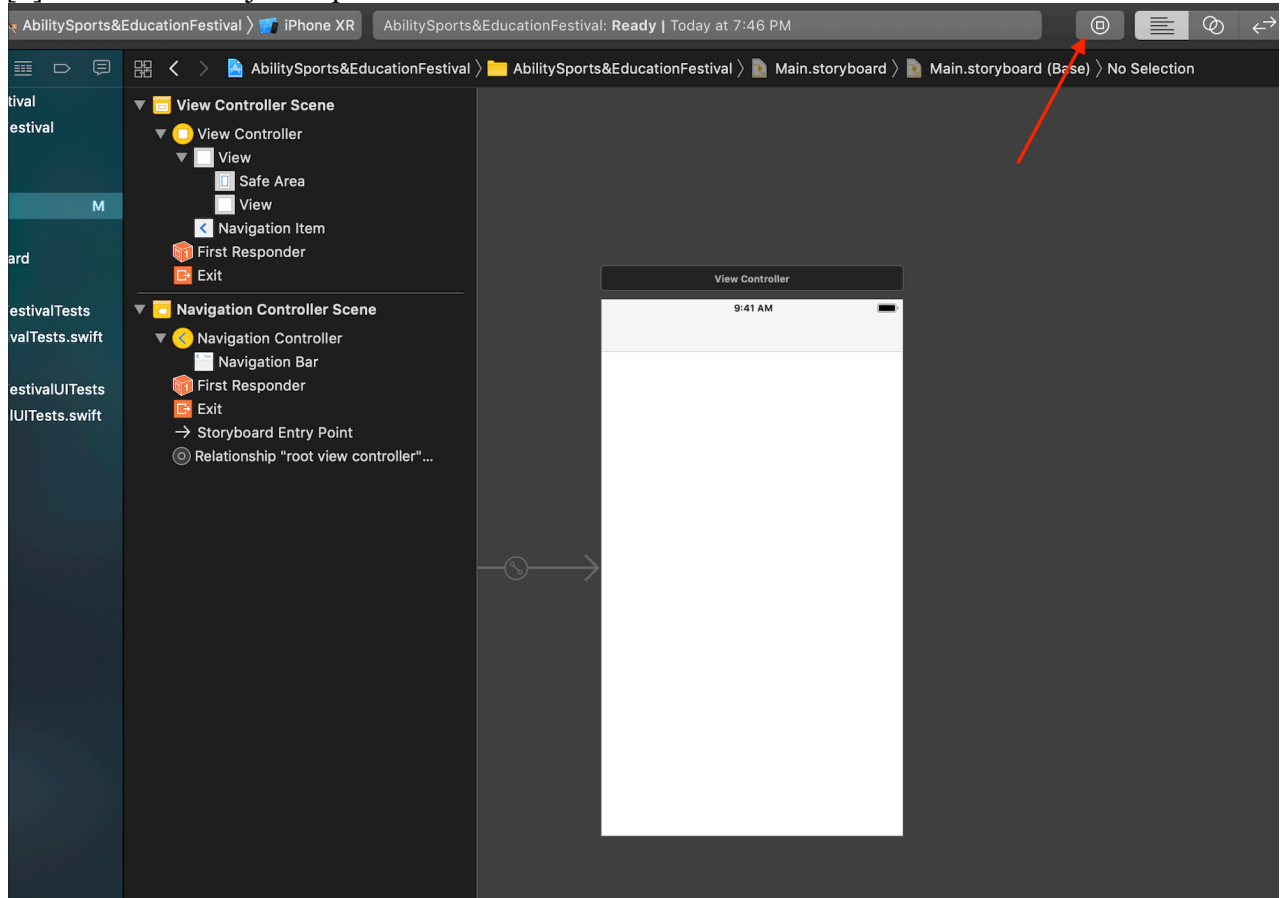


[37] Set the following settings for the outlets.

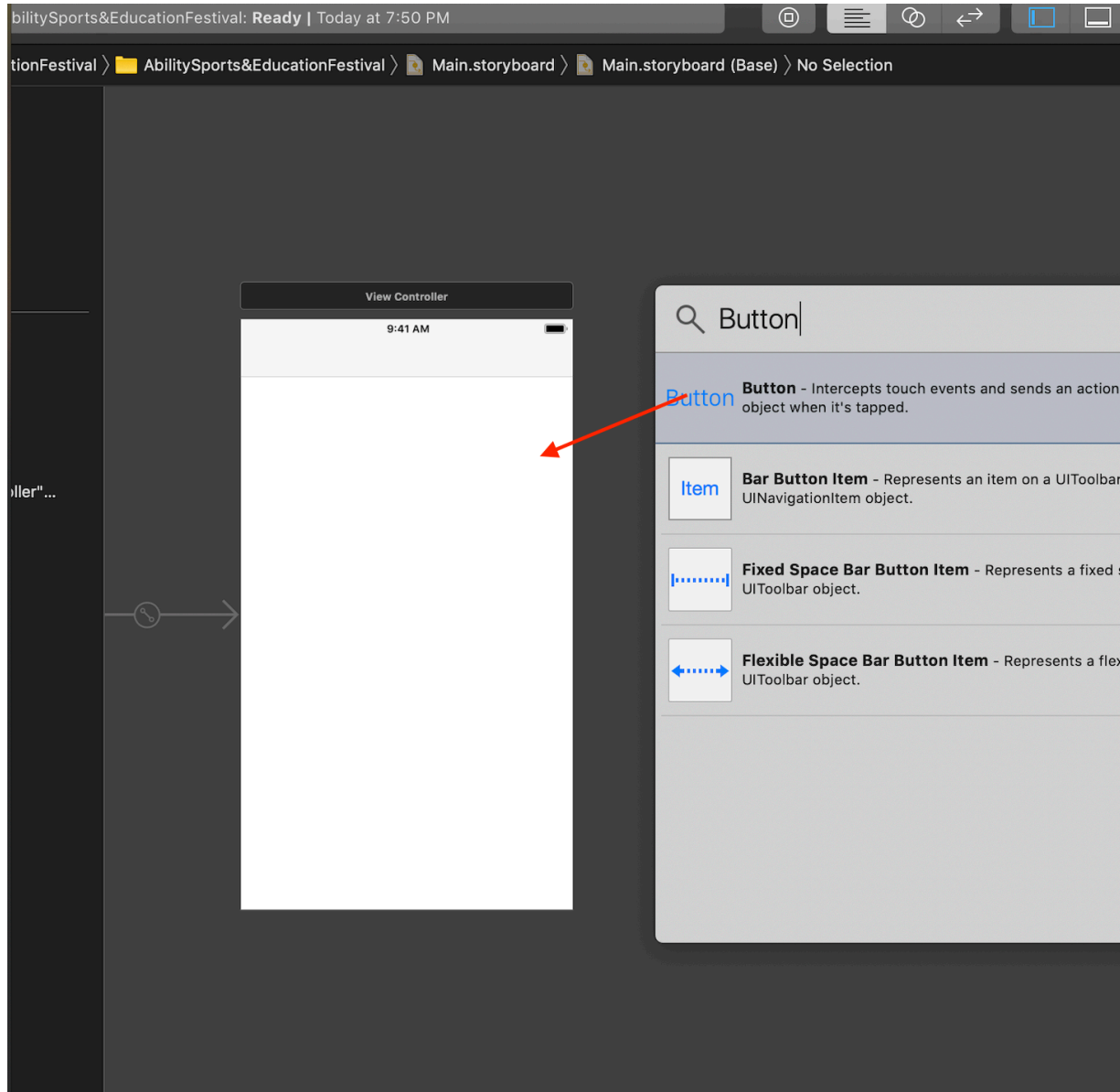


Create Home Page and Buttons

[1] Click on the object explorer

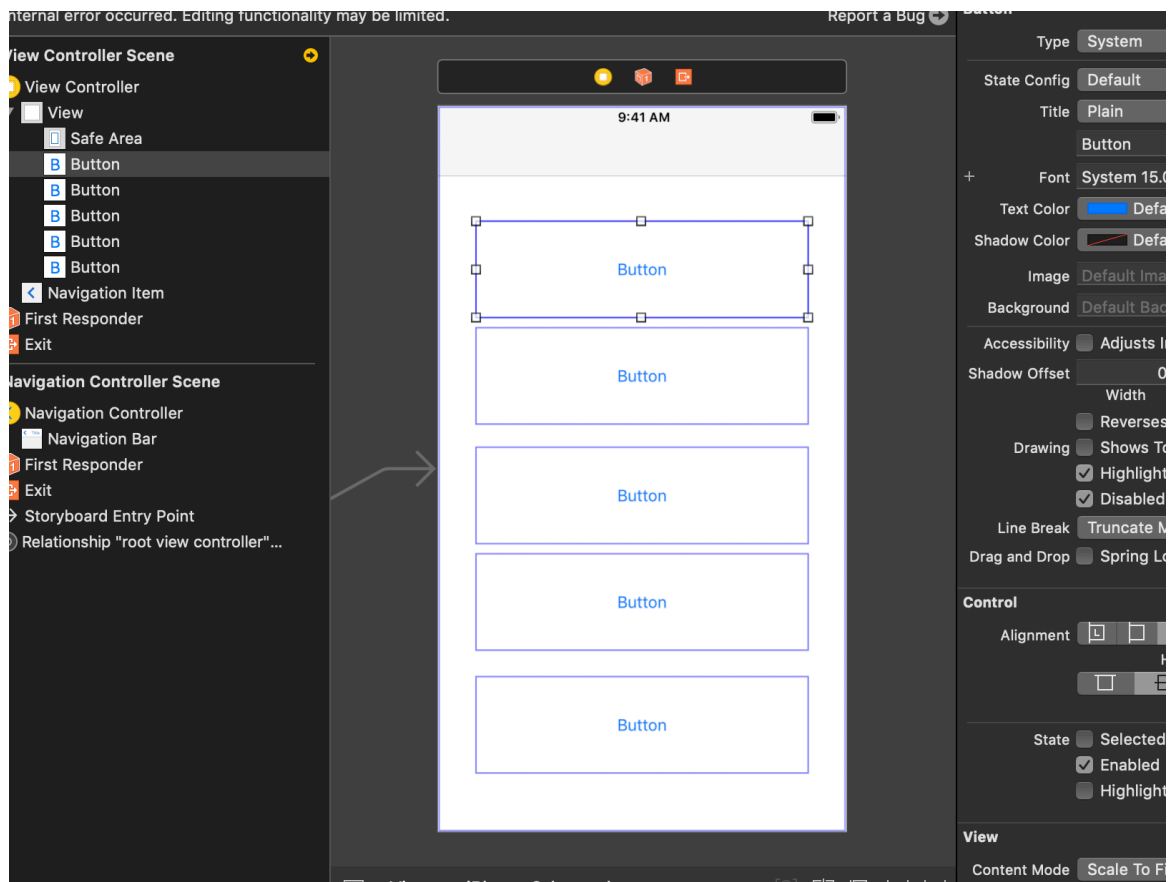


[2] Drag and place 5 buttons in the Main View. These will be the buttons that navigate the user to the other activities/views.



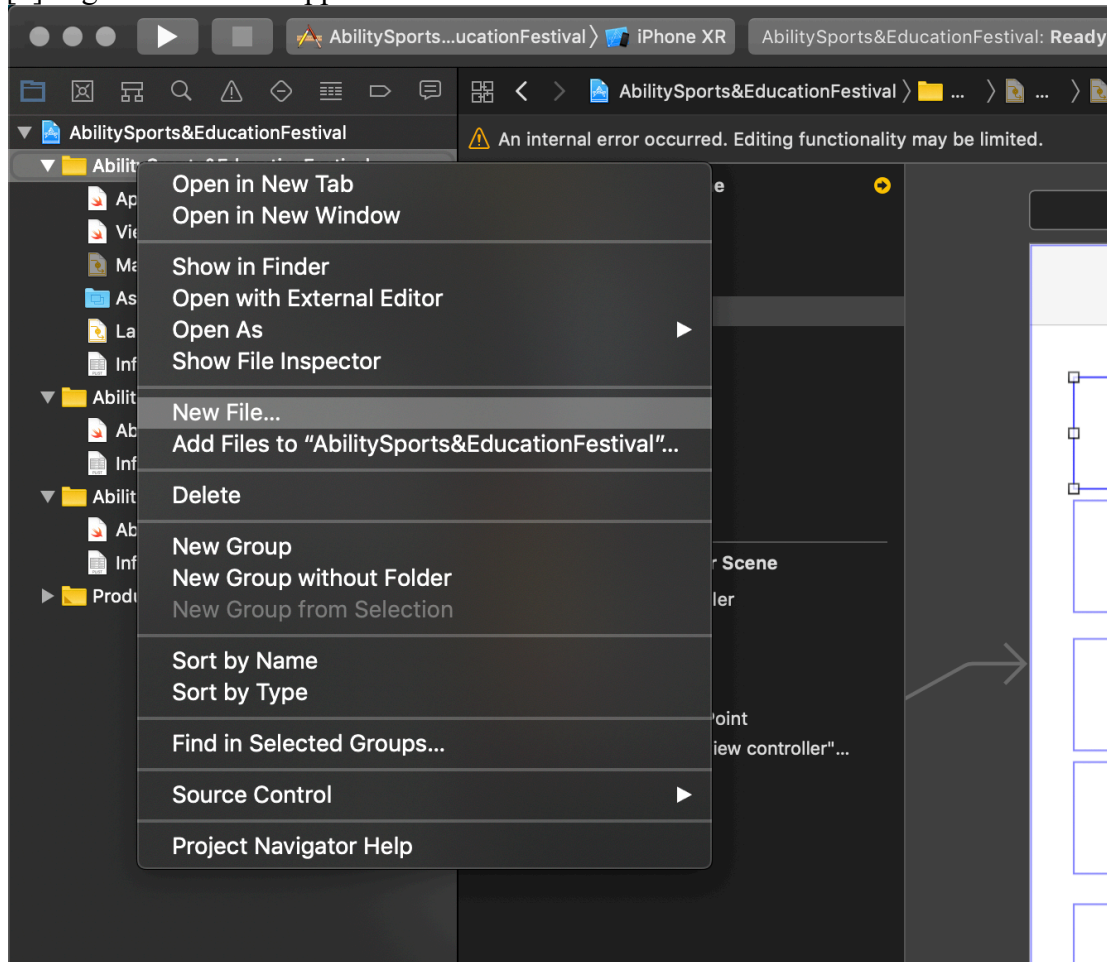
[3] After placing the buttons, the main story board should look like this:

Note: Make sure to add constraints.

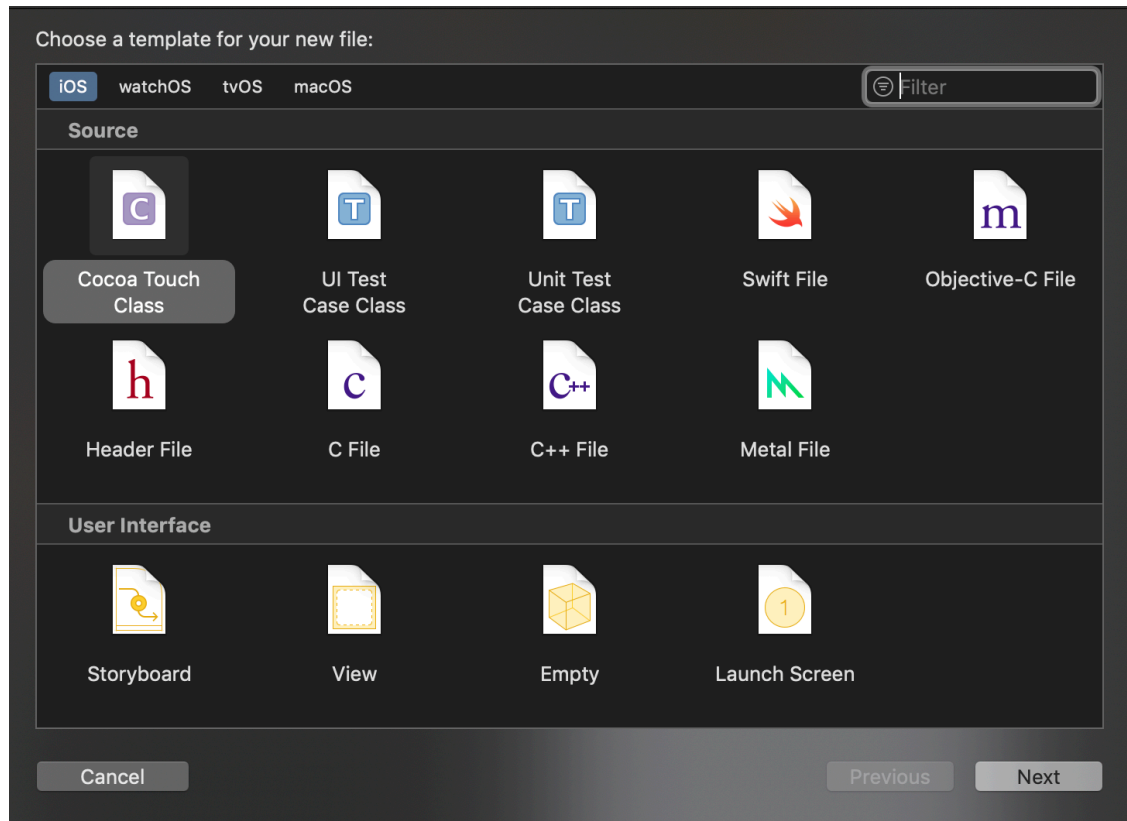


Create A new Class and Attach it to a new View

[1] Right-click on the applications main folder and click *New File*



[2] Select the *Cocoa Touch Class* and click *Next*.



[3] Name the new class file “*Events View Controller.*” This will be the view that users navigate to after pressing the “*Events*” button.

Choose options for your new file:

Class:

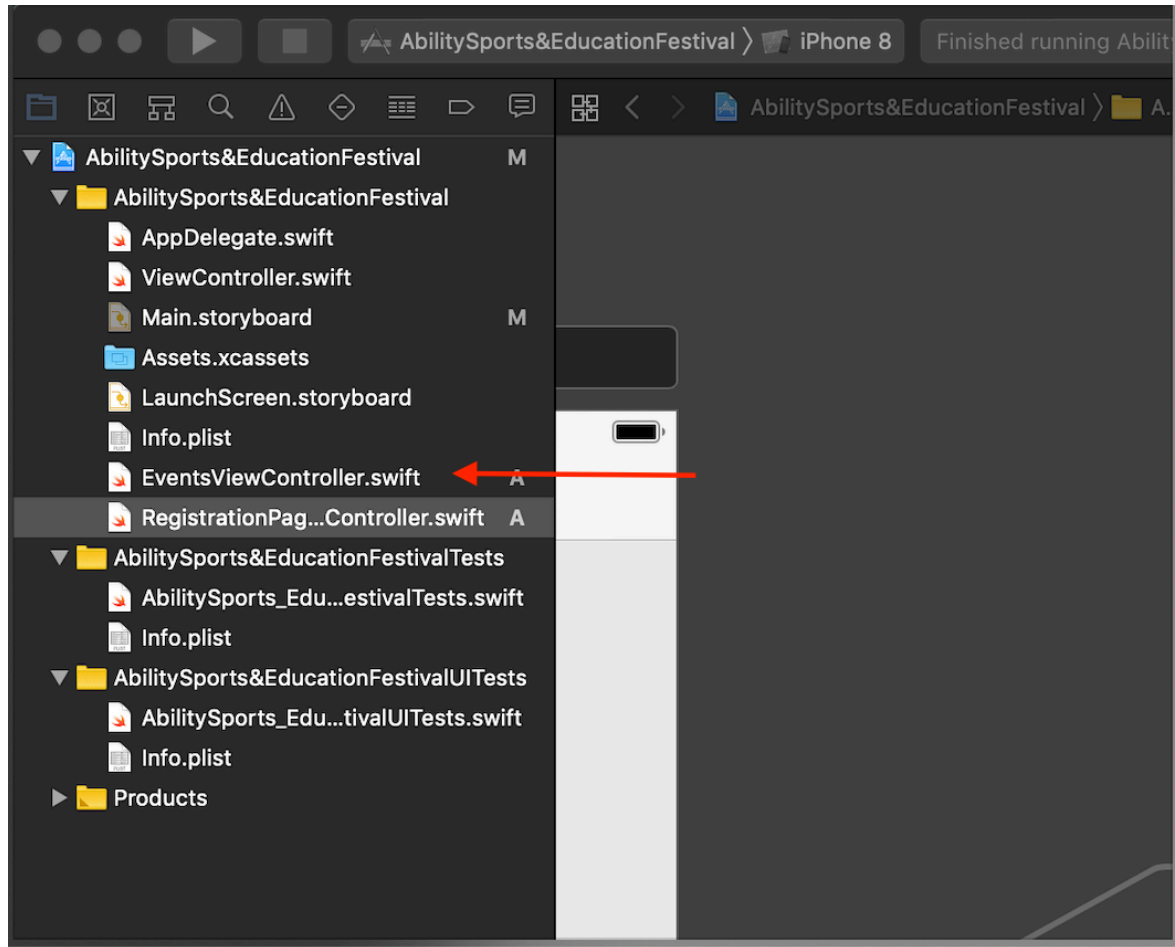
Subclass of: ▼

☐ Also create XIB file

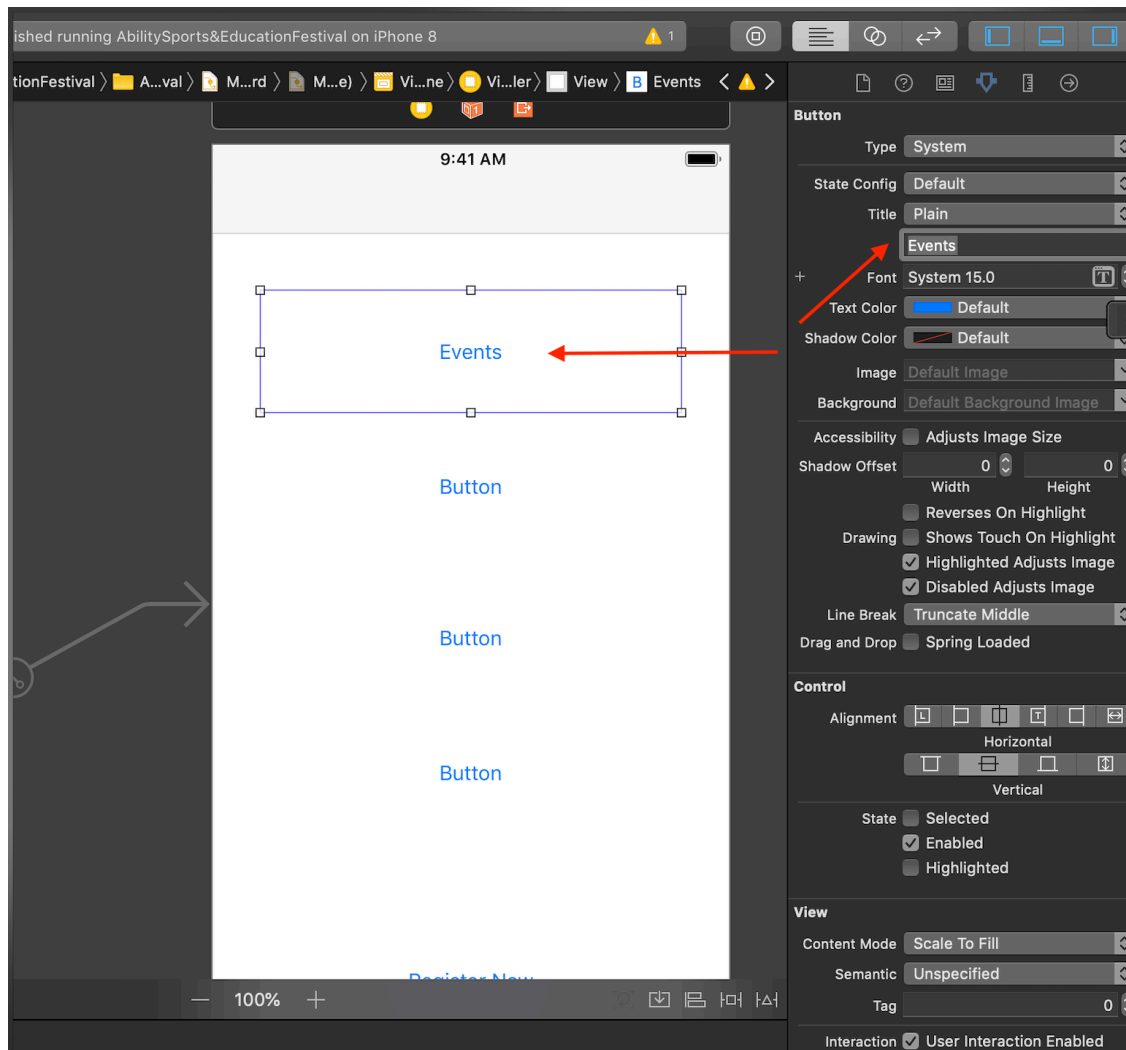
Language: ↕

Cancel Previous Next

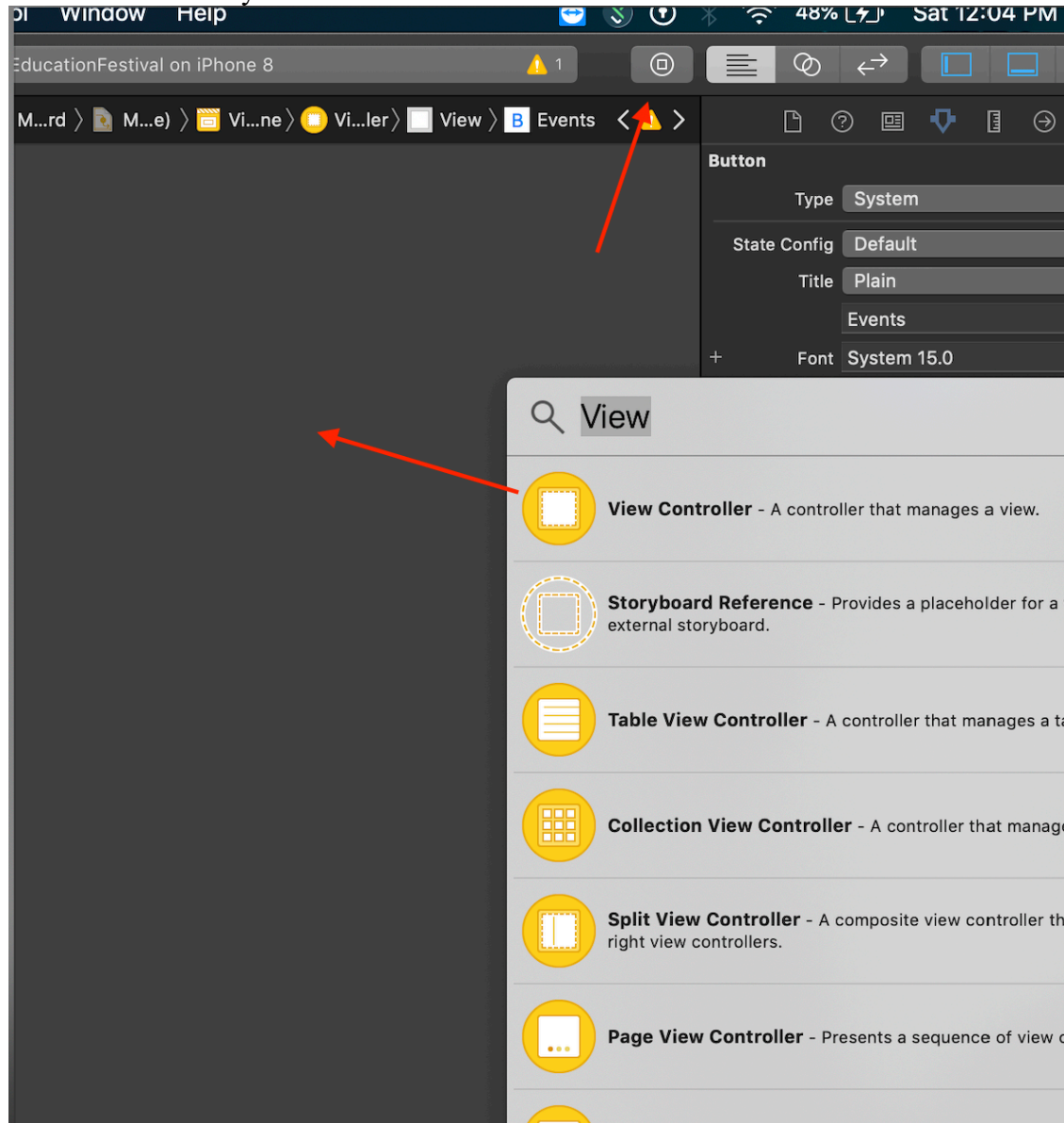
[4] Click *Next* and then *Create* to create the new file. You now be able to see the new *EventsViewController.swift* class in your project.



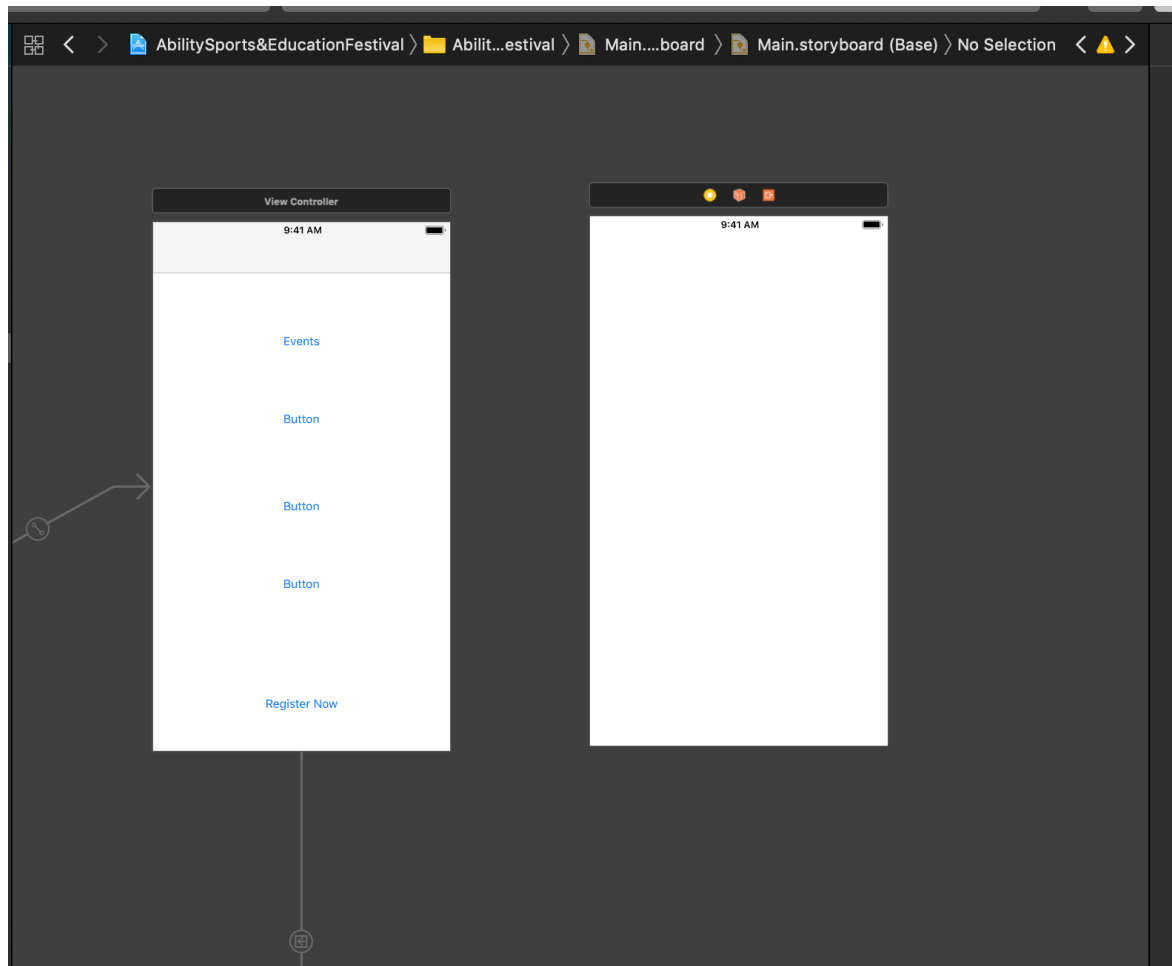
[5] Once the class has been created, locate and rename the first button of the application “Events.” Again, this will be the button that will navigate users to the Events View Controller.



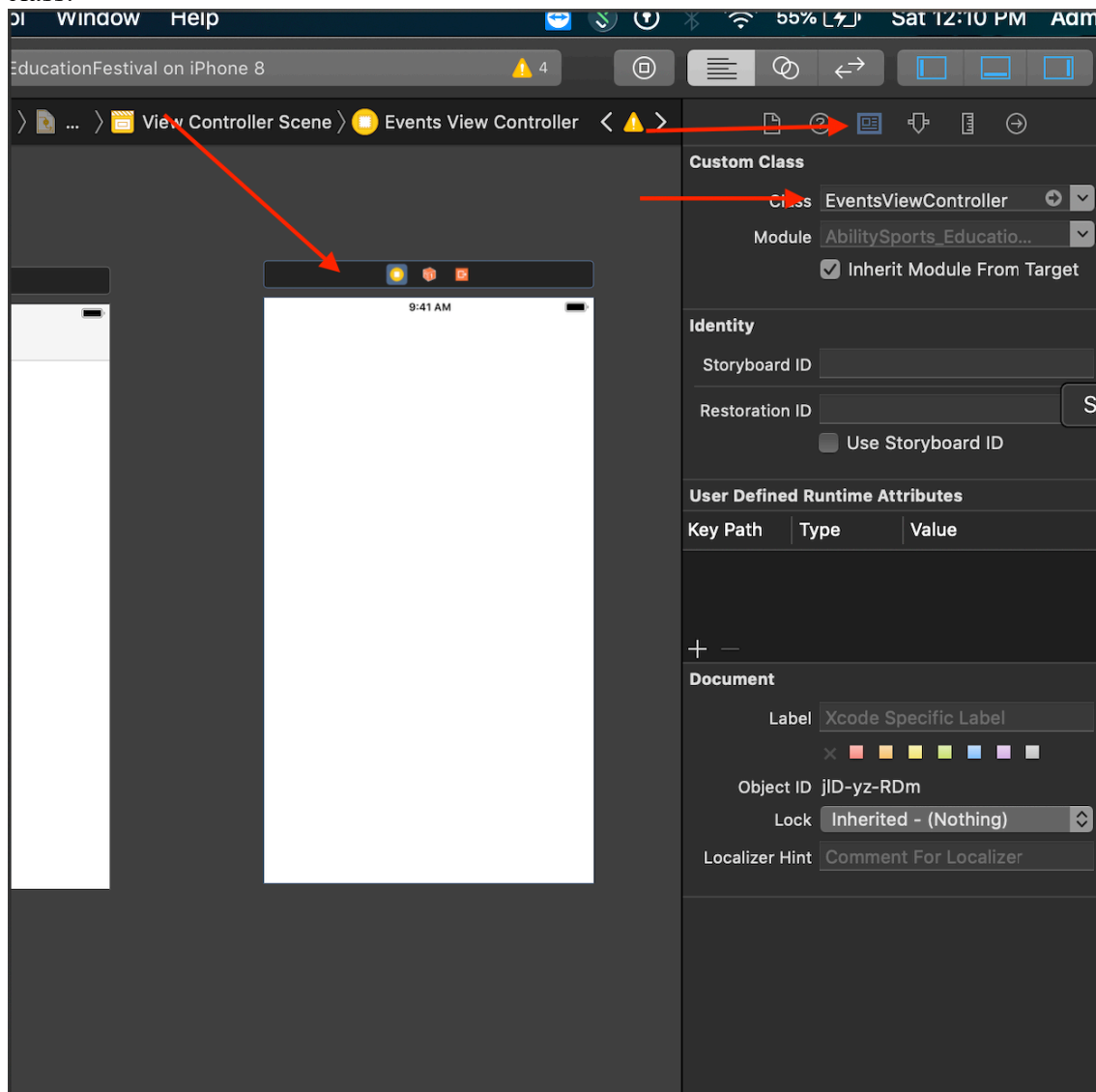
[6] Create a new Events View by clicking on the Objects Library and dragging a View Controller onto the Main.Storyboard



[7] You should now see two View Controllers.



[8] To connect the new View Controller to the EventsViewController class, click on the View controller. Then click on the *identity inspector*, and in the Class field drop down arrow, select the *EventsViewController.swift* class.



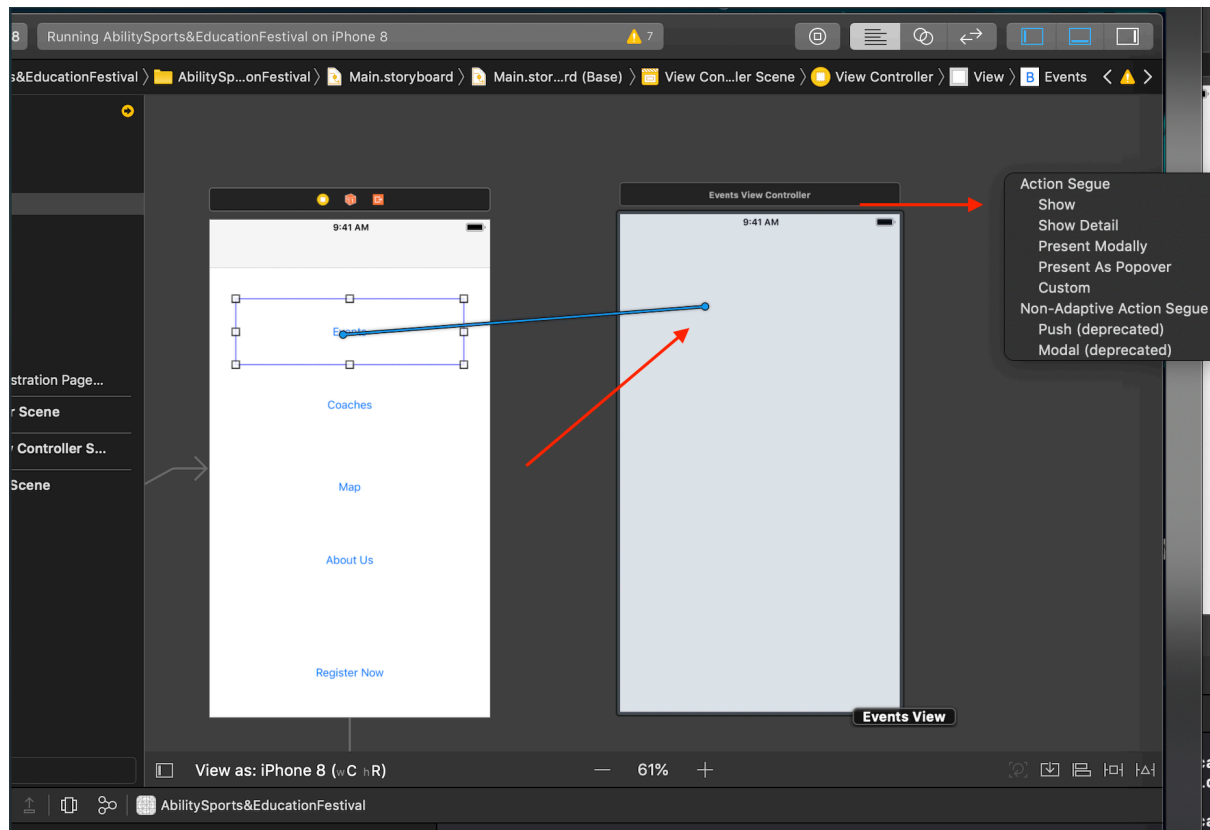
At this point the newly created View Controller will be associated with the EventsViewController.swift class.

Connect A Button to a View Controller using a Segue

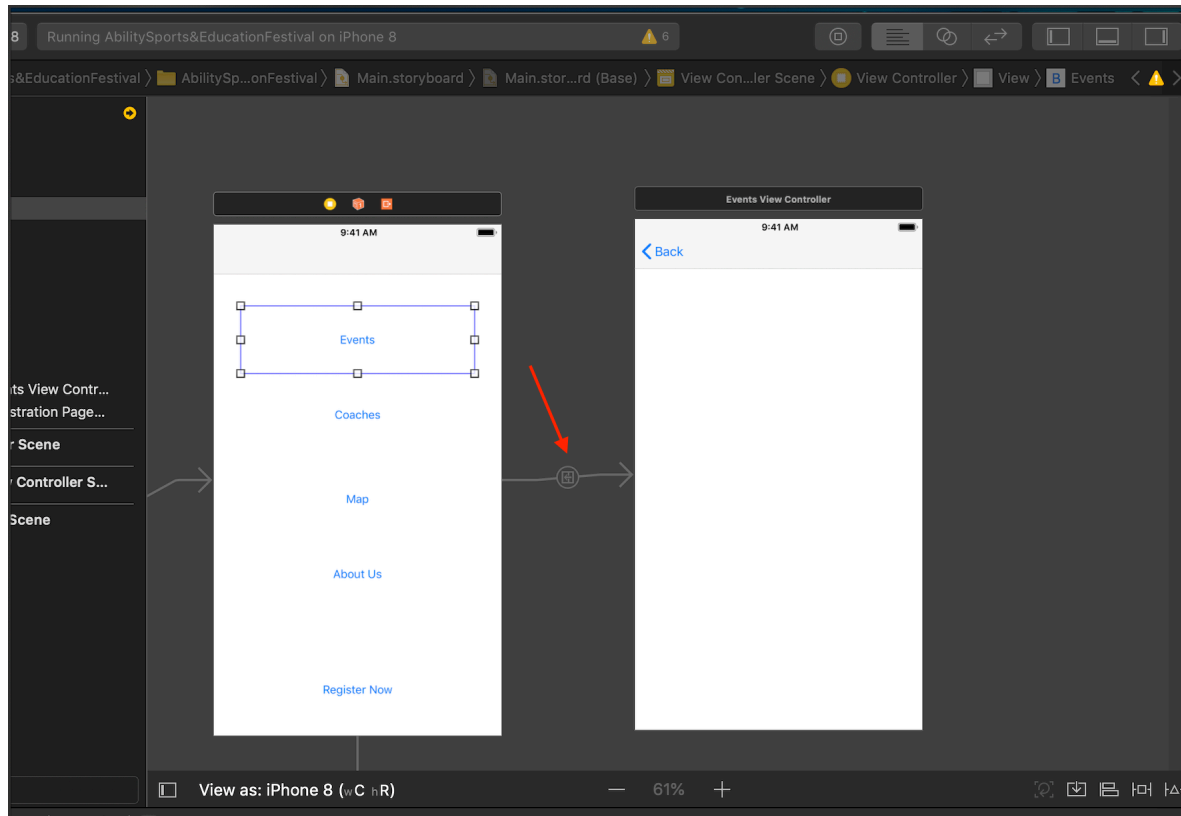
In the following steps, we will connect the Events button to the newly created View Controller using a segue. You can learn more about segues here:

<https://developer.apple.com/library/archive/featuredarticles/ViewControllerPGforiPhoneOS/UsingSegues.html>

[1] Now that we have created the Events view controller and assigned a class to it we will connect the Events button the Event View Controller. Open the Main.Storyboard. Right-click the events button and drag it to the Events View Controller. Click Show.



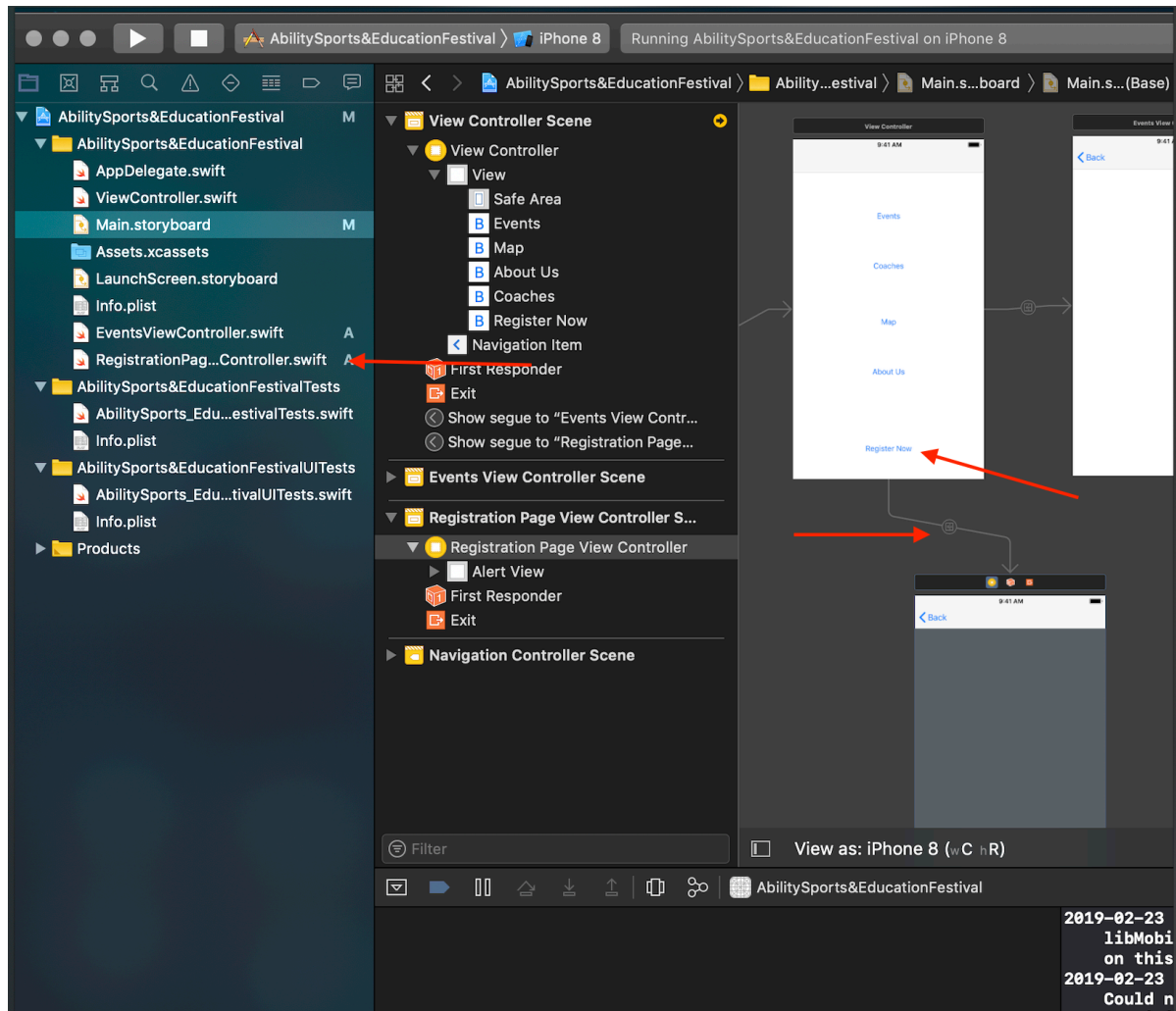
[2] Now, you should see a segue between both views. Start the app and you will see that pressing the Events button will navigate you the Events View Controller.



Create the Registration View and Connect it to the Ability Festival Online Registration Page

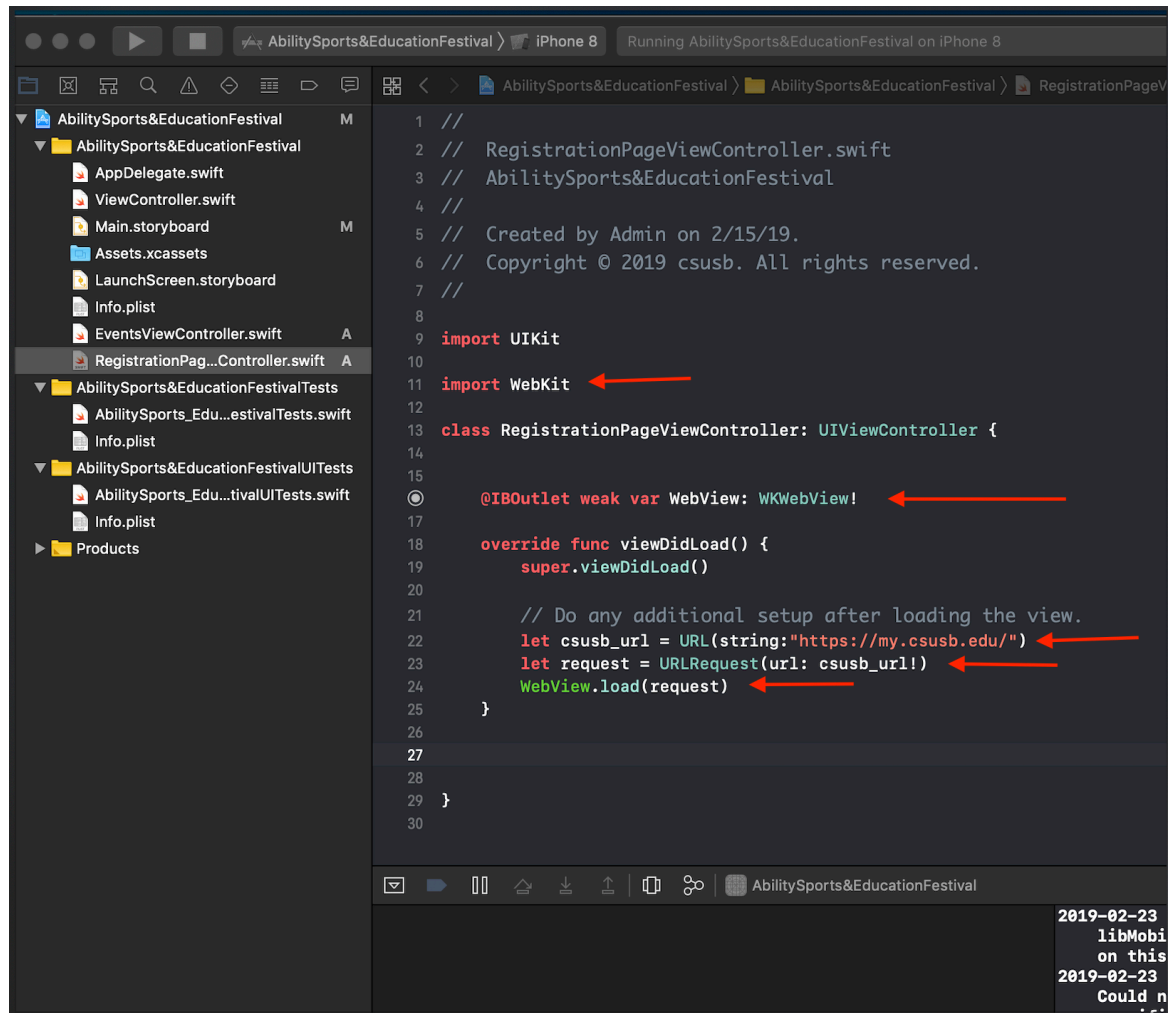
In the next steps, we will create a new view, class, and segue. The View will be called ***Register Now***, and the Class will be called ***RegistrationPageViewController.swift***.

[3] Create a new View Controller. Follow the instructions earlier in this document. Once you create a new view Controller, create a new class, connect the class to the view controller and create a segue from the “**Register Now**” button to the new **Register Now** View Controller. You should see something like this when done:



[4] Now that you have the registration view set up, we will need to write some code in the `RegistrationPageViewController.swift` class so that the View can display the schools online registration page for the event.

[5] Open the **RegistrationPageViewController.swift** class, and add the following Lines of Code.



[6] Here is the code highlighted:

```
//
//  RegistrationPageViewController.swift
//  AbilitySports&EducationFestival
//
//  Created by Admin on 2/15/19.
//  Copyright © 2019 CSUSB - Ability Sports & Education Festival. All rights reserved.
//
```

```
import UIKit
```

```
import WebKit
```

```
class RegistrationPageViewController: UIViewController {
```

```
@IBOutlet weak var WebView: WKWebView!
```

```
override func viewDidLoad() {  
    super.viewDidLoad()
```

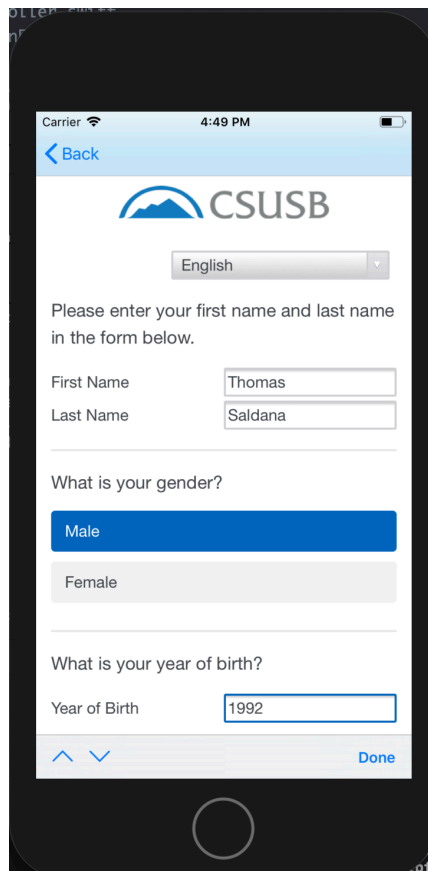
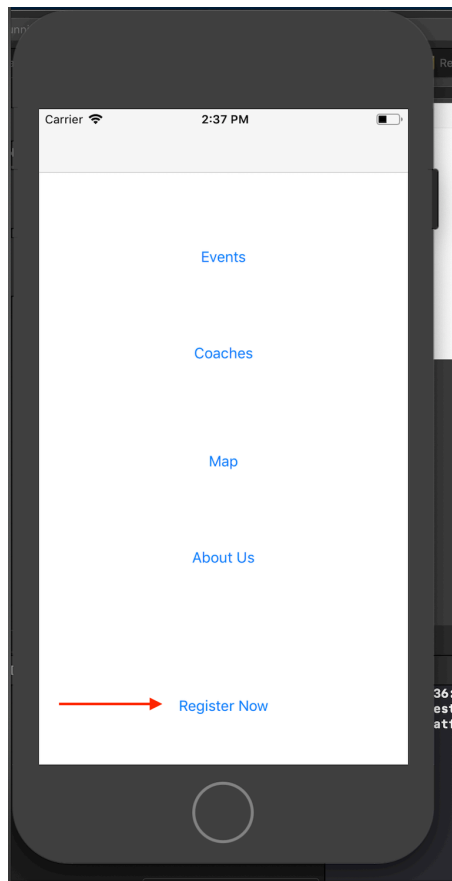
```
    // Do any additional setup after loading the view.
```

```
    let csusb_url = URL(string:"  
https://csusb.az1.qualtrics.com/jfe/form/SV_d530KmZzbuMkgM5 ")  
    let request = URLRequest(url: csusb_url!)  
    WebView.load(request)  
}
```

```
}
```

[7] Run the application. At this point, you should see the registration page when you click the **Register Now** button.

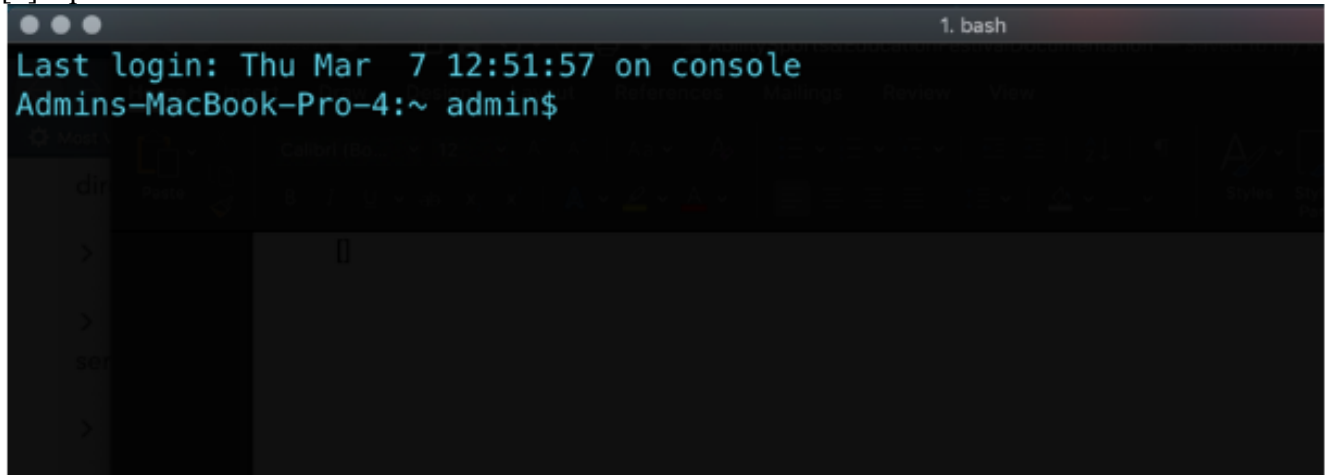
Ex)



Note: The registration page will point to the Ability Festival registration page on the day of the event.

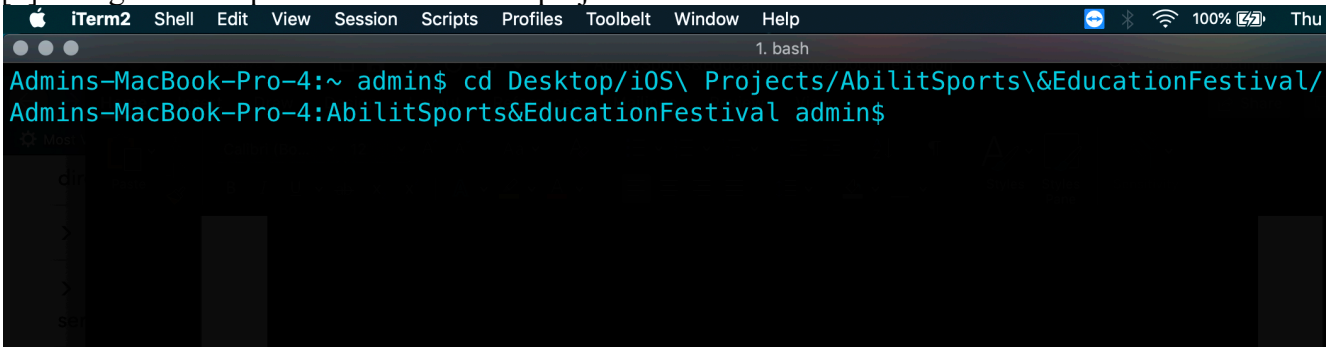
Create Initial ArcGIS Map

[1] Open a terminal window



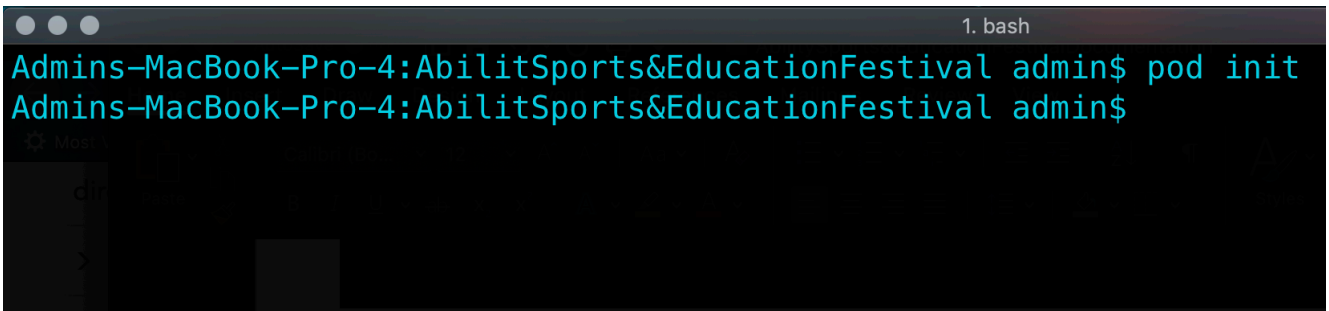
```
1. bash
Last login: Thu Mar  7 12:51:57 on console
Admins-MacBook-Pro-4:~ admin$
```

[2] Navigate to the path where the Xcode project is located at.



```
iTerm2  Shell  Edit  View  Session  Scripts  Profiles  Toolbelt  Window  Help
1. bash
Admins-MacBook-Pro-4:~ admin$ cd Desktop/iOS\ Projects/AbilitSports\&EducationFestival/
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$
```

[3] While in the directory where the current project is at, type *pod init*.



```
1. bash
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$ pod init
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$
```

[5] You should now see a file named *Podfile* under your working directory

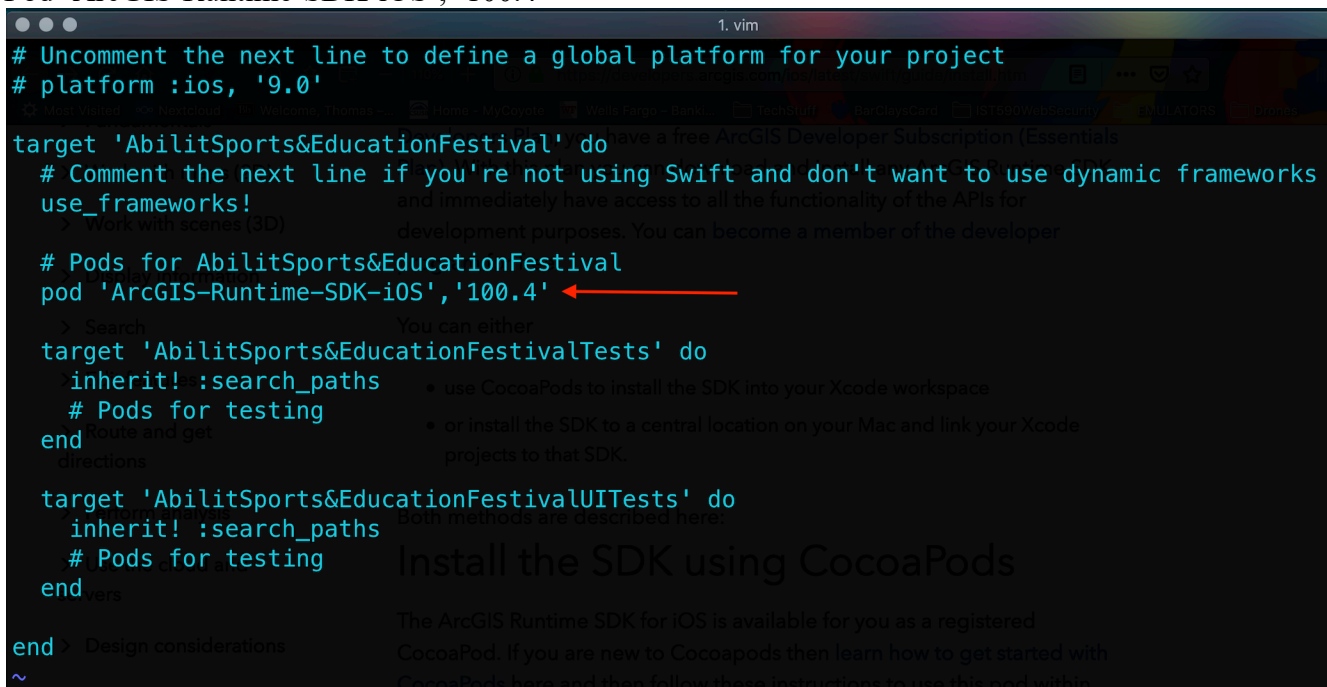
```
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$ ls -trl
total 8
drwxr-xr-x  4 admin  staff  128 Mar  2 10:34 AbilitSports&EducationFestivalTests
drwxr-xr-x  4 admin  staff  128 Mar  2 10:34 AbilitSports&EducationFestivalUITests
drwxr-xr-x@  5 admin  staff  160 Mar  7 13:55 AbilitSports&EducationFestival.xcodeproj
drwxr-xr-x 14 admin  staff  448 Mar  7 13:57 AbilitSports&EducationFestival
-rw-r--r--  1 admin  staff  515 Mar  7 14:06 Podfile
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$
```

[6] Vi into the file to open it

```
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$ vi Podfile
```

[7] Add the following line to the pod file:

Pod 'ArcGIS-Runtime-SDK-iOS', '100.4'



```
1. vim
# Uncomment the next line to define a global platform for your project
# platform :ios, '9.0'

target 'AbilitSports&EducationFestival' do
  # Comment the next line if you're not using Swift and don't want to use dynamic frameworks
  use_frameworks!

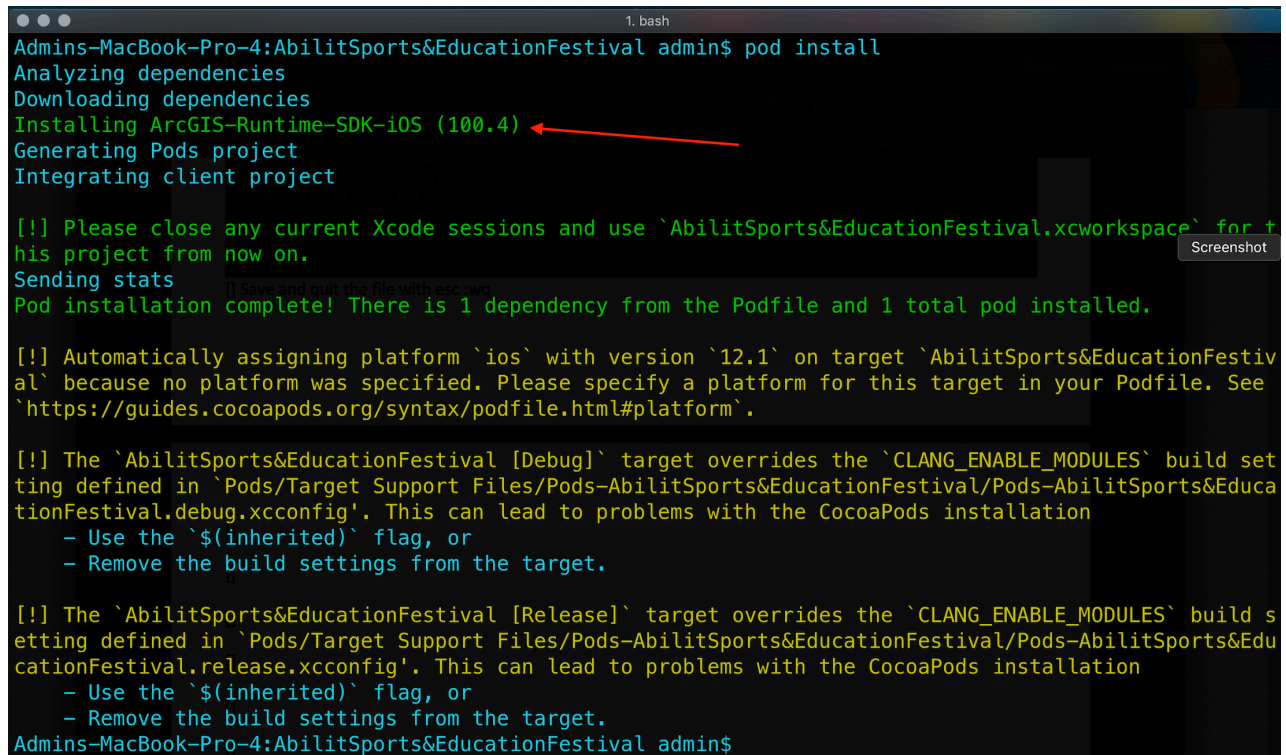
  # Pods for AbilitSports&EducationFestival
  pod 'ArcGIS-Runtime-SDK-iOS', '100.4'

  target 'AbilitSports&EducationFestivalTests' do
    inherit! :search_paths
    # Pods for testing
  end

  target 'AbilitSports&EducationFestivalUITests' do
    inherit! :search_paths
    # Pods for testing
  end
end
```

[8] Save and quit the file with esc :wq

[9] Type *pod install* to install the ArcGIS runtime sdk

A terminal window titled '1. bash' showing the output of the 'pod install' command. The output includes: 'Analyzing dependencies', 'Downloading dependencies', 'Installing ArcGIS-Runtime-SDK-iOS (100.4)' (highlighted with a red arrow), 'Generating Pods project', and 'Integrating client project'. It also shows a warning about Xcode sessions and a message that 1 dependency was installed. The prompt 'Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin\$' is visible at the bottom.

```
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$ pod install
Analyzing dependencies
Downloading dependencies
Installing ArcGIS-Runtime-SDK-iOS (100.4)
Generating Pods project
Integrating client project

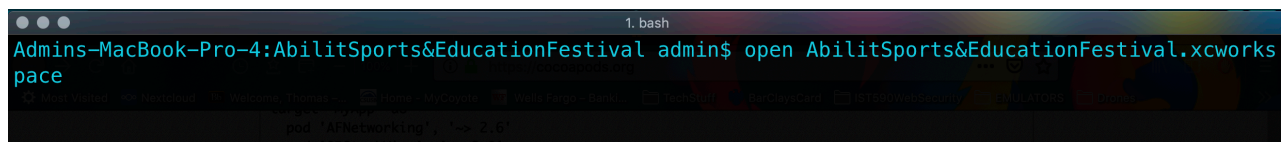
[!] Please close any current Xcode sessions and use `AbilitSports&EducationFestival.xcworkspace` for this project from now on.
Sending stats
Pod installation complete! There is 1 dependency from the Podfile and 1 total pod installed.

[!] Automatically assigning platform `ios` with version `12.1` on target `AbilitSports&EducationFestival` because no platform was specified. Please specify a platform for this target in your Podfile. See `https://guides.cocoapods.org/syntax/podfile.html#platform`.

[!] The `AbilitSports&EducationFestival [Debug]` target overrides the `CLANG_ENABLE_MODULES` build setting defined in `Pods/Target Support Files/Pods-AbilitSports&EducationFestival/Pods-AbilitSports&EducationFestival.debug.xcconfig`. This can lead to problems with the CocoaPods installation
  - Use the `$(inherited)` flag, or
  - Remove the build settings from the target.

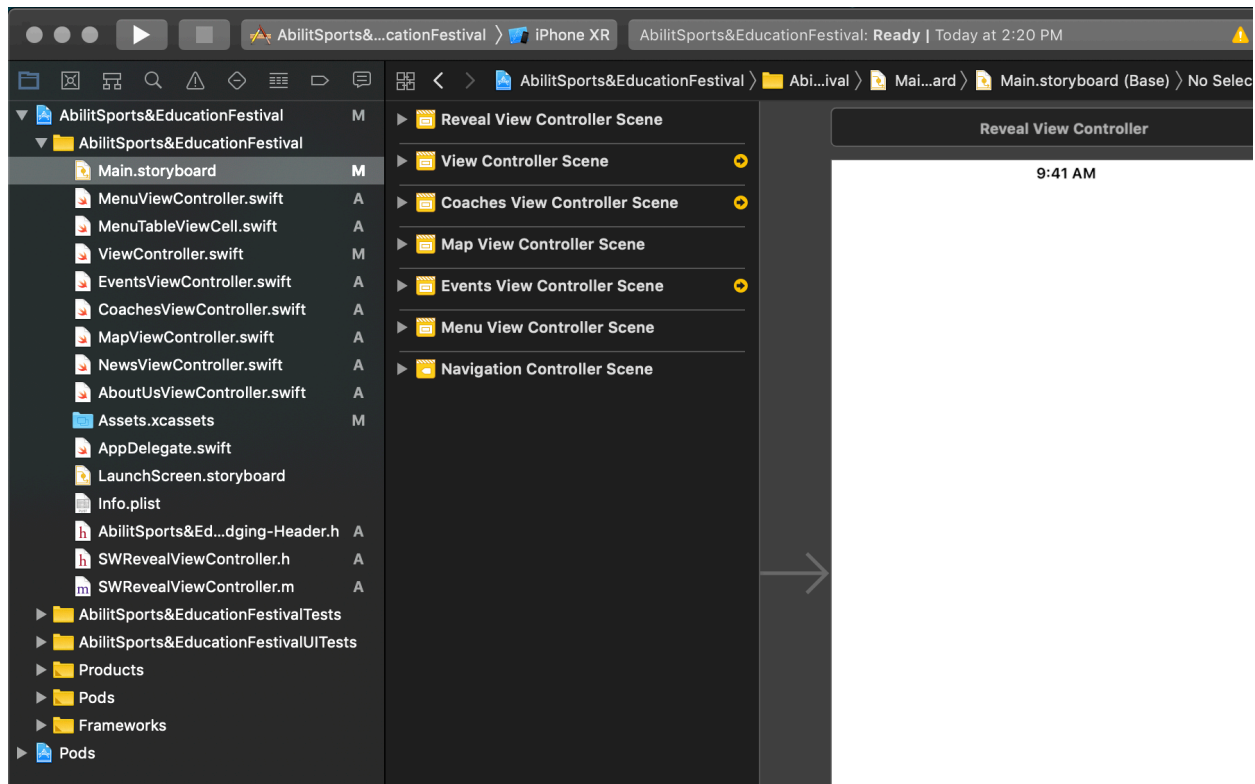
[!] The `AbilitSports&EducationFestival [Release]` target overrides the `CLANG_ENABLE_MODULES` build setting defined in `Pods/Target Support Files/Pods-AbilitSports&EducationFestival/Pods-AbilitSports&EducationFestival.release.xcconfig`. This can lead to problems with the CocoaPods installation
  - Use the `$(inherited)` flag, or
  - Remove the build settings from the target.
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$
```

[10] Open the newly created xcworkspace like mentioned in the above output.

A terminal window titled '1. bash' showing the command 'open AbilitSports&EducationFestival.xcworkspace' being entered. The prompt 'Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin\$' is visible at the bottom.

```
Admins-MacBook-Pro-4:AbilitSports&EducationFestival admin$ open AbilitSports&EducationFestival.xcworkspace
```

[11] A new Xcode Window should appear



[12] To learn more about cocoa pods, you can visit the following website:
<https://cocoapods.org/>

[13] Create a MapViewController.swift class and import the ArcGIS library

